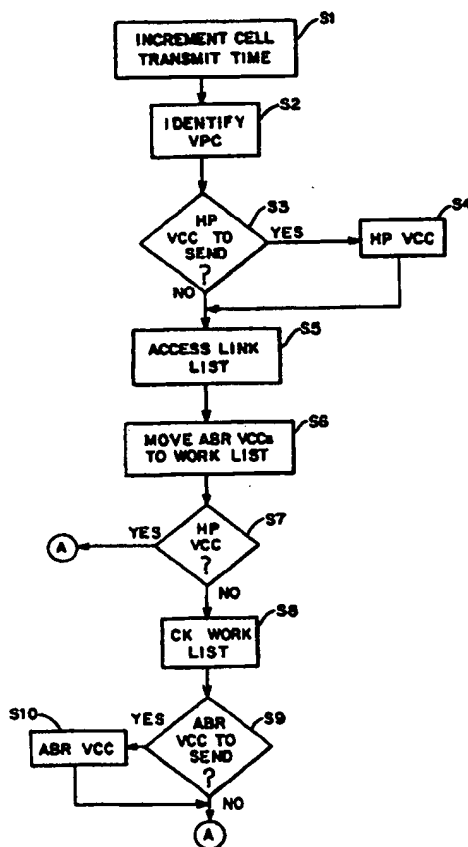


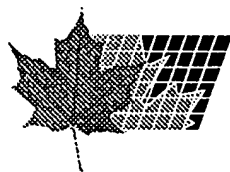
- (72) PEI, TONG-BI, JP
(72) ABE, KOHEI, JP
(71) CABLETRON SYSTEMS, INC., US
(71) KABUSHIKI KAISYA TOSHIBA, JP
(51) Int.Cl.⁶ H04L 12/56, H04Q 11/04
(30) 1997/11/18 (08/972,266) US
(54) **ORDONNANCEMENT HIERARCHIQUE POUR DIVERS TYPES
DE TRAFIC ATM**
(54) **HIERARCHICAL SCHEDULES FOR DIFFERENT ATM
TRAFFIC**



(57) L'invention concerne un ordonnanceur pour système ATM (mode de transfert asynchrone) qui intervient pour de multiples types de trafic (ex: CBR, VBR, ABR) avec dans chaque cas une pluralité de connexions de trajets virtuels (VPC) sur une liaison ATM. L'ordonnanceur fait appel à une table d'ordonnancement statique et gère une

(57) An ATM scheduler in accordance with the invention supports multiple transmission traffic types (e.g. CBR, VBR and ABR) for each of a plurality of virtual path connections (VPCs) on an ATM link. The scheduler uses a static scheduling table and maintains one or more dynamic scheduling tables. The scheduling





(21) (A1) **2,308,350**
(86) 1998/11/17
(87) 1999/05/27

ou plusieurs tables d'ordonnancement dynamique. La table statique hiérarchise les types de trafic et assure leurs ordonnancements respectifs pour chaque trajet virtuel. Pour chaque durée d'émission de cellule, l'ordonnanceur indexe une ligne de la table statique, attribuant par là même à un trajet virtuel la possibilité d'assurer l'émission à l'intérieur du créneau de la cellule correspondante. L'ordonnanceur examine un champ dans la ligne indexée de la table statique et recherche si une connexion pour circuit virtuel (VCC) avec service haute priorité a été effectuée au créneau actuel d'émission de la cellule, par exemple pour le trafic à débit binaire constant (CBR). Si tel est le cas, l'ordonnanceur donne à cette connexion la possibilité d'émettre. De même, pendant le traitement pour chaque créneau d'émission de cellule, l'ordonnanceur ATM prend un pointeur dans une liste de liaisons tirée de la liste statique et récupère cette liste dans la mémoire. Cette liste permet d'identifier au moins une connexion VCC de type service à débit binaire constant (ABR). Pendant chaque durée d'émission de cellule - au cas où le traitement relatif aux services hautement prioritaires n'a pas permis d'émettre la cellule dans le créneau imparti - l'ordonnanceur sélectionne l'une des connexions VCC de type service à débit binaire constant, ceci, soit directement dans la liste de liaison à laquelle il a actuellement accès, soit dans une liste de travail établie à partir de listes de liaison consultées au cours d'une série de créneaux d'émission de cellule. L'ordonnanceur lance alors l'émission sur le réseau d'une cellule pour le circuit ABR sélectionné dans le créneau de temps d'émission correspondant de la cellule.

table defines the hierarchy of traffic types and the scheduling for each type of traffic within each virtual path. In each cell transmit time, the scheduler indexes a line of the static table to identify one VPC assigned the opportunity to transmit in the respective cell transmit time. The scheduler reviews a field in the indexed line of the static table to determine if there is a high priority service virtual circuit connection (VCC) that has been assigned the current cell time slot, e.g. for constant bit rate (CBR) traffic. If so, then the scheduler provides the transmit opportunity for that VCC. Also, during processing for each cell transmit time, the ATM scheduler obtains a pointer to a link list from the dynamic table and retrieves the link list from memory. The link list identifies at least one available bit rate (ABR) service type VCC. During each respective cell transmit time, if the processing relating to the high priority service(s) did not result in cell transmission, the scheduler selects one of the identified ABR service type VCCs, either directly from the currently accessed link list or from a work list compiled from link lists accessed over a series of cell transmit times. The scheduler initiates transmission of a cell for the selected ABR circuit over the link in the respective cell transmit time.



PCTWORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

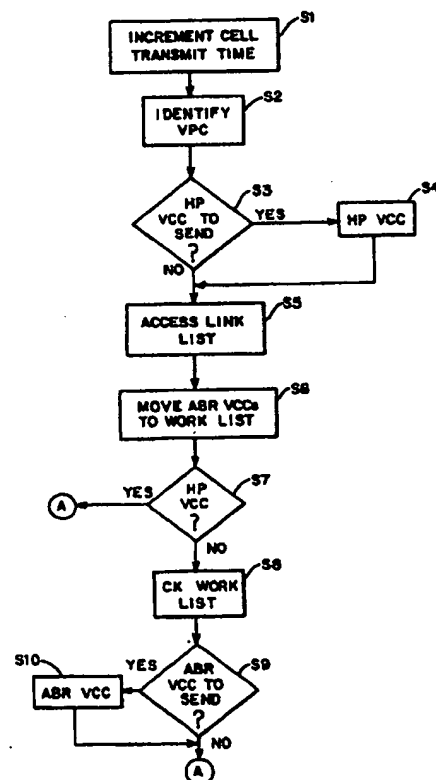
INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁶ : H04L 12/00	A2	(11) International Publication Number: WO 99/26378 (43) International Publication Date: 27 May 1999 (27.05.99)
<p>(21) International Application Number: PCT/US98/24541</p> <p>(22) International Filing Date: 17 November 1998 (17.11.98)</p> <p>(30) Priority Data: 08/972,266 18 November 1997 (18.11.97) US</p> <p>(71) Applicants: CABLETRON SYSTEMS, INC. [US/US]; 35 Industrial Way, Rochester, NH 03867 (US). KABUSHIKI KAISYA TOSHIBA [JP/JP]; Semiconductor Intellectual Property & Licensing Dept., 580-1, Korikawa-cho, Saiwai-ku, Kawasaki-shi 210 (JP).</p> <p>(72) Inventors: PEI, Tong-Bi; 2296 Cascade Street, Milpitas, CA 95035 (US). ABE, Kohel; 38-1, Tsukuno-cho, Tsurumi-ku, Yokohama (JP).</p> <p>(74) Agent: HENDRICKS, Therese, A.; Wolf, Greenfield & Sacks, P.C., 600 Atlantic Avenue, Boston, MA 02210 (US).</p>		<p>(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).</p> <p>Published <i>Without international search report and to be republished upon receipt of that report.</i></p>

(54) Title: **HIERARCHICAL SCHEDULES FOR DIFFERENT ATM TRAFFIC**

(57) Abstract

An ATM scheduler in accordance with the invention supports multiple transmission traffic types (e.g. CBR, VBR and ABR) for each of a plurality of virtual path connections (VPCs) on an ATM link. The scheduler uses a static scheduling table and maintains one or more dynamic scheduling tables. The scheduling table defines the hierarchy of traffic types and the scheduling for each type of traffic within each virtual path. In each cell transmit time, the scheduler indexes a line of the static table to identify one VPC assigned the opportunity to transmit in the respective cell transmit time. The scheduler reviews a field in the indexed line of the static table to determine if there is a high priority service virtual circuit connection (VCC) that has been assigned the current cell time slot, e.g. for constant bit rate (CBR) traffic. If so, then the scheduler provides the transmit opportunity for that VCC. Also, during processing for each cell transmit time, the ATM scheduler obtains a pointer to a link list from the dynamic table and retrieves the link list from memory. The link list identifies at least one available bit rate (ABR) service type VCC. During each respective cell transmit time, if the processing relating to the high priority service(s) did not result in cell transmission, the scheduler selects one of the identified ABR service type VCCs, either directly from the currently accessed link list or from a work list compiled from link lists accessed over a series of cell transmit times. The scheduler initiates transmission of a cell for the selected ABR circuit over the link in the respective cell transmit time.



- 1 -

**HIERARCHICAL SCHEDULES FOR
DIFFERENT ATM TRAFFIC****Technical Field**

5 The present invention relates to techniques and devices for scheduling asynchronous transfer mode (ATM) traffic, for different bit rate services assigned to virtual circuit connections within a plurality of virtual path connections, for transmission over an ATM link.

Background Art

Modern society is increasingly dependent on the ability to communicate information.
10 More and more applications require communications of varying quantities of information between users. The trend in communications technology is to develop packet or cell based systems for communications transport and switching at ever higher speeds.

Many services having different requirements drove the development of separate networks. For example, analog voice telephone services utilize a complex network of voice traffic switches,
15 lines and trunks to provide ubiquitous switched voice connectivity virtually throughout the world. The modern telephone network carries most voice traffic in digitized form, typically using time division multiplexing techniques. The switched voice network can carry some data traffic, using modems or ISDN interfaces. However, the telephone network cannot readily switch higher speed data traffic, therefore a variety of separate data networks evolved. Examples of such data
20 networks include X.25, frame relay and SMDS. The construction, operation, maintenance and upgrading of such disparate networks for different services are increasingly complex and expensive, particularly as traffic demands continue to increase.

Asynchronous transfer mode (ATM) transport, an advanced, high-speed packet switching technology, has emerged as the latest form of packet or cell based switching. ATM promises fast
25 cell switching for wide ranges of traffic demands. In ATM, information is organized into cells having a fixed length and format. Each cell includes a header, primarily for identifying cells relating to the same virtual connection, and an information field or "payload".

The ATM standard defines a cell size of 53 bytes or octets. The first five bytes of each cell form a header, and the remaining 48 bytes represent payload data. The header of each cell
30 includes a field for a virtual path identifier (VPI) and a Virtual circuit identifier (VCI), to identify the particular communication to which each cell relates.

ATM is intended to carry virtually any type of information that can be expressed in or

- 2 -

converted to digital form, from voice telephone traffic, to real-time video, to high-speed file transfers, to faster than real-time video, etc. ATM based networks are eliminating the need for different networks to carry different types of traffic. In ATM, transfer is asynchronous in the sense that the recurrence of cells that contain information from any particular sender is not necessarily periodic. Each device using the ATM network submits a cell for transfer when it has a cell to send, not when they have an assigned or available transmission time slot. Once scheduled and aggregated, the ATM cells ride in synchronous slots on a high-speed media, such as a SONET optical fiber. ATM allows any arbitrary information transfer rate up to the maximum supported by the ATM network, simply by transmitting cells more often as more bandwidth is needed.

Different types of communication require different transport rates. Also, different communications require different levels of service quality, referred to as quality of service or QoS. For example, real-time video transmission requires a high constant bit rate to maintain synchronism, whereas packet data communications do not.

Although the ATM standard specifies both virtual path and virtual circuit switching, devices currently available utilize only virtual circuit connection or VCC based routing. For example, a user network interface (UNI) is known which provides constant bit rate (CBR) and available bit rate (ABR) services using a single virtual path connection (VPC) identifier. CBR service takes precedence over ABR traffic. Each time that there is an opportunity for a CBR circuit to transmit, the interface transmits a cell for that circuit. If the circuit presents no data to send, the interface sends cells from the circuits assigned ABR service if there is such a circuit whose service time is due at this time.

The various circuits are differentiated as different virtual circuit connections (VCCs) having different VCC identifiers, both internally in the interface and in the headers of the transmitted cells. The user network interface runs a VCC based scheduler, to schedule transmission of cells for the various services. To provide CBR service, the scheduler maintains a CBR table. This table includes a VCC identifier for each circuit subscribing to CBR service in the order that service is scheduled for the respective circuits. Constant bit rate service takes precedence over available bit rate service. A table pointer traverses the CBR table at the link cell rate. Each time that the pointer points to a listing in the table that contains the VCC identifier for a circuit subscribing to constant bit rate service, the user network interface will transmit one cell from that circuit. Thus, each CBR circuit receives a percentage of the CBR link bandwidth

11 03 11 03

proportional to the number of listings for the circuit in the table divided by the total number of listings in the table. For example, if VC1 appears in every other slot in the table (half the slots of the table), then the user network interface schedules an opportunity for that VCC to transmit a cell every other cell transmit time. The virtual circuit connection VC1 has the opportunity to use half of the link bandwidth.

Each time that the scheduler within the user network interface accesses a listing or slot in the CBR, the scheduler also accesses the next listing or slot in an ABR table. Thus, when the scheduler reads a listing from the CBR table, it also reads a listing from the ABR table. Each listing in the ABR table comprises head and tail link list pointers. These link list pointers point to the beginning and end of a link list of VCC identifiers for circuits subscribing to available bit rate service. The scheduler moves the accessed link list of VCC identifiers to a work list. In this manner, the scheduler accumulates a work list of ABR circuit identifiers as the interface runs through cell times of the ATM link. Whenever a cell transmit opportunity is not used by a constant bit rate service, the scheduler ideally would go through the accumulated work list until it finds a virtual circuit having a cell to send and enables transmission of the cell from that circuit.

The CBR table is static. The ABR table, however, is dynamic. If the processing through the ABR work list enables an ABR service type circuit to actually transmit a cell, then the scheduler removes the served circuit VCC from the work list. This VC is put back on the ABR table, but at a specifically selected point further down the table. The selection of that point in the table effectively reschedules the service for the sequence of listed ABR circuits. The distance that the listing is moved down the table depends on the type of flow control algorithm in use.

The prior art system assumed all traffic was within one virtual path, for example identified by a virtual path identifier (VPI) of 0. The ATM link may be viewed as a pipe, running from the interface to the next node of the ATM network. The CBR and ABR table approach provides constant bit rate service and available bit rate service, but the use of one VP or VPC for all such services limits the ability to provide such services to a large number of circuits through a single interface. All of the service is within the one pipe defined by the one VPC. The use of a single VPC limits the ability to segregate the bandwidth of the ATM pipe from the user network interface to the next node of the ATM network. The ATM pipe can carry cells from only a certain number of circuits and can support only so many CBR circuits. Also, the interface supports only the two types of services and cannot easily support those services in combination with other services, such as variable bit rate (VBR).

AMENDED

In the U.S. Patent No. 5,619,502, Kahn discloses a communication subsystem including a static scheduler and a dynamic scheduler. The static scheduler accesses a scheduling list that specifies either a virtual circuit or a dynamic scheduling indication for each cell slot on a communication link. The static scheduler selects the virtual circuit specified by the scheduling list if the scheduling list specifies the virtual circuit for the cell slot and the dynamic scheduler selects a virtual circuit from a dynamic scheduling list for transfer of the outbound communication cell if the scheduling list specifies the dynamic scheduling indication for the cell slot. The communication subsystem also includes a counter that counts if an idle communication cell is transferred over the communication link due to an underrun such that the virtual circuit specified by the dynamic scheduling list is skipped if the counter indicates the underrun.

In European Patent Application No. EP 0 700 186 A1, Banks discloses a method and apparatus for regulating virtual-channel cell transmission. In order to schedule the transmission onto a link of cells associated with different virtual channels, a transmission schedule is set up associating virtual channels with respective transmission time slots. Each time a cell is sent, the corresponding virtual channel is re-allocated to a time slot that is offset from the current one by an amount dependent on the desired rate of transmission. This desired rate is specified for each virtual channel in a corresponding transmission rate indicator (69) comprising a fixed integer part (70), a fixed fractional part (71), and a variable accumulation part (72). The fixed parts (70, 71) of the indicator define the desired rate whilst the accumulation part (72) is used in calculating each successive offset. More particularly, to determine the offset of the time slot next to be allocated to a virtual channel, the three part (70-72) of the corresponding transmission rate indicator (69) are added up and the integer component (74) of the result is used as the offset, the fractional remainder (75) being stored back to the accumulation part (72) of the transmission rate indicator. This arrangement permits a fine control to be achieved on the average transmission rate. A similar approach can be used to control the rate of credit generation in systems where transmit credits are used to regulate cell transmission.

In European Patent Application EP 0 717 532 A1, Calvignac discloses dynamic fair queuing to support best effort traffic in an ATM network. A traffic control apparatus implemented in a virtual path ATM communication system including a plurality of hops interconnected by physical links which comprise virtual paths including a plurality of virtual channels. A connection between two hops is defined as the combination of a physical link, a

AMENDED SHEET

3B

bandwidth and a best effort service. ATM data cells conveyed on said best effort service are routed from hop to hop by analyzing their virtual connection identifier. This is done by managing a pool of free queues being able to store ATM data cells, each of said queues storing all the incoming ATM data cells having the same virtual channel identifier.

AMENDED BY 107

- 4 -

Disclosure of the Invention

The present invention solves the above discussed problem with the prior art by scheduling ATM cell transport over a link using both virtual path connections and virtual circuit connections. The hierarchical scheduler in accord with the invention facilitates CBR, VBR and ABR services, within logically separate virtual paths or pipes, carried over the same ATM link. The scheduler uses one static table for assigning cell transmit time slots to the virtual paths and for identifying virtual circuits that may use those slots for relatively high priority services, such as CBR and VBR. The scheduler uses and updates one or more dynamic tables for scheduling lower priority traffic, for example for ABR service.

Thus, in one aspect, the present invention relates to a method of scheduling asynchronous transfer mode (ATM) cell traffic, for transmission over a link in a plurality of cell transmit times. The link supports traffic for a number of virtual path connections, thus each virtual path connection may utilize bandwidth on the link. In each respective cell transmit time, one virtual path connection is identified as having been assigned the respective cell transmit time as its cell transmit opportunity, from a line of the static scheduling table indexed by cell transmit time. Each virtual path connection supports high priority traffic, such as constant bit rate traffic or variable bit rate traffic, for a plurality of virtual circuit connections. The ATM processing device performing the scheduling also determines from the indexed line if there is a high priority service virtual circuit connection, from among those associated with the identified virtual path connection, that is assigned the respective cell transmit time. If the scheduling device identifies such a virtual circuit connection, then the scheduler provides the opportunity to transmit a cell for that high priority service via the virtual circuit on the link in the respective cell transmit time.

If the high priority service is a CBR service and the corresponding virtual circuit connection has data to send, the connection carries a transmitted cell over the link in the respective cell transmit time. If the service is a variable bit rate service, the scheduler checks whether the circuit is ready to send. If so, the scheduler enables transmission in this cell time.

Also, during processing for each cell transmit time, the ATM processing device performing the scheduling indexes a line of a dynamic scheduling table. From data in that line of the dynamic table, the scheduler identifies at least one virtual circuit connection having available bit rate service. For example, the scheduler preferably obtains a pointer from the line of the dynamic table. The scheduler uses the pointer to retrieve one of a number of link lists from memory. The link list identifies one or more virtual circuit connections having the ABR service.

- 5 -

The indexing of the dynamic table associates the identified virtual circuit connection(s) with the respective cell transmit time. The preferred embodiment utilizes one dynamic table for each virtual path. For each cell transmit time, the scheduler uses the dynamic table for the virtual path indexed in the static table. The dynamic table points to link lists of available bit rate service type virtual circuit connections associated with the one virtual path connection.

If the processing relating to the high priority service did not result in cell transmission in the time slot, the ATM processing device performing the scheduling selects an ABR circuit identified in one of the link lists for transmission. Typically, when the device accesses a link list, the device adds one or more of the identified virtual circuit connection(s) to a work list. During a respective cell transmit time in which there is no high priority traffic scheduled, the scheduling device examines the work list. Specifically, the device accesses the work list to identify an available bit rate service type virtual circuit connection for which there is a cell to transmit, and the cell for that circuit is transmitted over the link in the respective cell transmit time.

The present invention also encompasses ATM processing devices that incorporate elements for performing the scheduling operation described above.

For example, in one embodiment, the invention encompasses an ATM processing device, including an ATM cell transmitter, a memory and a programmable cell transmission scheduler. The scheduler controls the scheduling of cells for transmission over a communication link. A static scheduling table is stored in memory, for use by the programmable cell transmission scheduler. The static scheduling table includes a plurality of lines, each of which contains an index identifying one of a plurality of virtual path connections that may utilize the link during a respective cell transmit time. A field in each line of the static scheduling table may contain an identifier of a virtual circuit connection having a high priority service (e.g., constant bit rate service). In the preferred implementation of this embodiment, each line of this table also includes a field for an identifier of a virtual circuit connection having a somewhat lower priority service, e.g. variable bit rate service.

The memory of the ATM device also stores link lists, each of which identifies one or more virtual circuit connections. The ATM device further comprises at least one dynamic table stored in memory. Each line of the dynamic table has a field containing a pointer. The pointer points to one of the virtual circuit connection link lists for one or more low priority services, e.g. connections having available bit rate service.

Additional objects, advantages and novel features of the invention will be set forth in part

in the description which follows, and in part will become apparent to those skilled in the art upon examination of the following or may be learned by practice of the invention. The objects and advantages of the invention may be realized and attained by means of the instrumentalities and combinations particularly pointed out in the appended claims.

5 Brief Description of the Drawings

Figures 1A and 1B together form a high level flow diagram of ATM cell transmission scheduling in accord with the present invention.

Figure 2 is a block diagram of a data communication system including an ATM segmentation and reassembly circuit, performing ATM cell transmission in accord with the
10 present invention.

Figure 3 is a high level block diagram of the ATM segmentation and reassembly circuit.

Figure 4 is a more detailed block diagram of the ATM segmentation and reassembly circuit.

Figure 5 is an example of a first implementation of a scheduling table useful in the present
15 invention.

Figure 6 illustrates a static scheduling table used in a second embodiment of the invention.

Figures 7A to 7D depict simplified portions of dynamic scheduling tables used for scheduling ABR type ATM cell transmission in accord with the second embodiment of the present invention.

Figure 8 is a process diagram useful in understanding work list operations for ABR
20 service

Figure 9 is a flow chart illustrating the process flow in accord with the second embodiment of the present invention.

Best Mode for Carrying out the Invention

25 The present invention enables the scheduling of asynchronous transfer mode (ATM) cell traffic for transmission over a link. ATM transport devices operating in accord with the present invention incorporate a traffic scheduler which segregates traffic on the link into a plurality of virtual path connections, and within each virtual path, the scheduler segregates traffic into a plurality of virtual circuit connections. The scheduler utilizes one or more tables to assign traffic
30 of a variety of types into respective cell transmit time slots. Figures 1A and 1B form a simplified flow chart illustration of the schedule processing of the present invention.

AMENDED SHEET

- 7 -

The scheduler increments cell transmit time (step S1). A key feature of the present invention relates to use of multiple virtual path connections (VPCs). For each respective cell transmit time, the scheduler therefore identifies one virtual path connection that is assigned to the cell transmit time (step S2). The division of the ATM link into separate virtual paths (VPs) using multiple virtual path identifiers (VPIs) and assigning time slots to the VPCs allows the division of traffic on the link into several logical paths or pipes. The overall (physical) link then can support CBR traffic and ABR traffic within each logical pipe. The new system also supports variable bit rate (VBR) traffic.

The scheduler supports high priority services, such as CBR service, for a number of virtual circuit connections (VCCs) within each virtual path connection (VPC). The scheduler therefore determines if there is a high priority VCC, associated with the identified VPC, for which there is a cell to send (step S3). If there is no cell ready to send for this CBR, the opportunity is passed to low priority VCC associated with this VPC. For a variable bit rate (VBR) service type circuit, the scheduler will identify the VCC and will check to see if there is a cell from the source device ready for transmission over that VCC. If there is a high priority cell to send, the scheduler registers the VCC identifier for that high priority circuit and sets an appropriate status bit (step S4), and processing flows to step S5. If there is no high priority circuit with a cell to send in the present transmit opportunity, then the schedule processing advances directly from step S3 to step S5. If steps S3, S4 identified a high priority VCC having a cell to transmit, then the scheduler will initiate transmission of the cell for that virtual circuit connection at the end of one cycle through the process flow (in step S16), as discussed more below.

In step S5, the scheduler accesses a link list. The link list is one associated with the VPC identified in step S2. The link list identifies one or more virtual circuit connections having available bit rate service (ABR) whose service time is due. Each virtual path connection (VPC) supports ABR services for multiple virtual circuit connections (VCCs). As discussed more below, the preferred embodiments utilize one or more tables to associate the ABR VCCs with their respective VPCs and with cell transmit time opportunities.

The scheduler may transfer all of the VCCs from the one accessed link list to a work list, or the scheduler may check the status of the circuits on the link list and transfer only the VCC identifiers for those ABR circuits with a cell ready to send from the accessed link list to the work list (step S6). Over a series of cell transmit times, the scheduler will transfer individual VCC identifiers from a number of link lists associated with a number of VPCs to build up one or more

work lists of ABR circuits awaiting an available slot in which to transmit a cell. Thus, during any cell transmit time for which the scheduler has no higher priority cell scheduled, it then fetches the ABR VCC from the head of the work list for the VPC. If this ABR VCC has a cell to send, the cell is sent over the link. Otherwise, a null cell is sent over the link. In the simplified flow diagram illustrated in Figure 1A, this is implemented through a check of the ready status for the high priority circuits (step S7) followed by work list processing (steps S8-S10).

If the scheduler found a VCC with a high priority (HP) cell to send, step S7 branches to step S11. However, if the scheduler has found no high priority VCC with a cell to transmit, process flow branches from step S7 to step S8. At this point, the scheduler checks the work list to set the ABR VCC from the head of the work list. If this VCC has a cell awaiting transmission, processing flows through step S9 to step S10, in which the scheduler registers the VCC identifier for that ABR service circuit and sets an appropriate status signal. If this VCC has no cell ready to transmit or the work list is empty, step S9 branches the process flow to bypass step S10.

In step S11 the scheduler determines which cell to send. If there is a high priority circuit for which there is a cell ready to send, processing branches to step S12 in which the scheduler loads the cell from the source to the ATM transmitter. If there is an ABR type circuit for which there is a cell ready to send, processing branches to step S13 in which the scheduler loads the cell from the particular ABR source to the ATM transmitter. In step S14, the scheduler moves the identifier for the ABR type VCC granted the transmit opportunity from the work list. If a null cell is sent for this VCC, this VCC is not put back to the schedule table. Otherwise, this VCC is rescheduled to a new location for future processing, as will be discussed in more detail with regard to the preferred embodiments. If there was no high priority circuit and no ABR circuit with a cell ready to send, processing branches to step S15 in which the scheduler loads a null cell to the ATM transmitter.

In step S16, the scheduler causes the ATM transmitter to send the loaded cell. Thus, if the high priority processing identified a VCC with a right to transmit during this cell transmit time and having a cell ready to transmit (S3, S4), then the scheduler loaded a cell for that CBR type VCC (S12). The ATM transmitter sends the cell over the VCC on the link having the high priority service in step S16. Alternatively if the work list process identified an ABR type VCC with a cell ready to send (S8, S10), then the scheduler loaded a cell from that ABR type VCC (S13), and the ATM transmitter sends the ABR service cell over that VCC on the link in step S16. If processing did not identify either a high priority VCC or an ABR VCC with a cell to send, then

AMENDED SHEET

the scheduler loaded a null cell (S15), and the transmitter sends the null cell over the link in step S16. It should be noted that the cells are transmitted on the identified virtual path connection (VPC) on the link. In ATM, virtual path connections and virtual circuit connections are made up of cells in the transmit stream that contain the VPIs and VCIs of those connections in the headers
5 of the actual cells.

After transmission, processing returns through step S17 to step S1, where the scheduler increments the cell transmit time. The scheduler will go through the process again, but in a typical sequence, the next pass through the scheduler process will identify a different VPC and a VCC within that VPC. In this manner, repeated flow through the process will allocate cell
10 transmit time slots to a variety of high priority and available bit rate services within the different VPCs. The hierarchical scheduler therefore supports different types of traffic within each logical path or pipe identified as a VPC.

The above-discussed operations are examples of typical operations. The traffic scheduling principles of the invention may facilitate other types of ABR processing. For example, if the
15 scheduler selects a VCC from a link list only if the circuit has a cell to send, in a time slot in which there is no high priority service scheduled and the work list is otherwise empty, the scheduler can enable transmission over the circuit identified from the link list without moving the VCC to the work list.

The ATM traffic scheduling of the present invention may apply in a variety of ATM
20 processing devices, such as ATM edge devices and other ATM routers and/or switches. The preferred implementation of the present invention applies the scheduling principles in a segmentation and reassembly circuit, typically used in an ATM user network interface (UNI). A high level functional description of a network and a UNI containing the scheduler is presented followed by a discussion of the preferred embodiments of the scheduling process implemented
25 in the segmentation and reassembly circuit.

Figure 2 illustrates a user network interface (UNI) 10 providing a two-way data communication connection between a local area network (LAN) 11 and a high speed ATM link 17. The ATM link 17 typically connects to a switch or router of an ATM network. The LAN 11 provides data communications between various data devices connected thereto. In the simple
30 example illustrated, the LAN 11 connects to a plurality of PCs 13 and to one or more servers 15. The UNI 10 and the connection thereof through the LAN 11 enables the PCs 13 and/or the server 15 to send and receive data communications over assigned virtual circuits on the ATM link 17.

AMENDED SHEET

As discussed in more detail later, the segmentation and reassembly circuit 23 assigns virtual channel connections (VCCs) to data communication services from the PCs 13 and the server 15 and schedules transmissions depending on the types of service and bandwidths allocated to each of those data devices.

The ATM segmentation and reassembly circuit 23 controls the data flow between the cell based virtual circuits assigned on the ATM link and the packet based virtual circuits through the LAN 11 to the data devices 13, 15. The circuit 23 segments and adapts outgoing data for ATM transport and schedules transmission in assigned virtual paths and circuits. As part of this operation, the circuit 23 inserts VPIs and VCIs in the cell headers to logically place cells in the virtual paths and circuits. The circuit 23 also reassembles payload data from ATM cells into packets for transport to the data devices. The circuit 23 performs all necessary address administration in both directions.

Operations of the ATM segmentation and reassembly circuit 23 are controlled by a microprocessor 27. The microprocessor 27 serves as the node control processor for the UNI 10, for example to administer service negotiation. The ATM segmentation and reassembly circuit 23 also connects to a memory 29, such as a static random access memory (SRAM). The circuit 23 stores scheduling tables, link lists and work lists in the memory 29. Assembled outgoing cells containing segmented data and VPI/VCI values are buffered and queued in the memory.

The LAN 11 carries data communications in some standard type of packet data protocol, such as Ethernet. The packets include media access control layer addressing information, to facilitate two-way communication over the LAN. The UNI 10 includes a LAN interface 21 for physical connection to the LAN 11 and for conforming information going to and from the UNI 10 to the particular LAN protocol. The LAN interface provides a two-way MAC level connection or interface to the ATM segmentation and reassembly circuit 23.

The link 17 carries ATM cells in some high speed transport format. For example, the link may be a DS3 communication channel on electrical cable or an OC-1 or OC-3 on optical fibers. The ATM segmentation and reassembly circuit 23 connects through its physical interface port to a link interface 25. The link interface 25 conforms the ATM cell information going to and from the UNI 10 to the signal type (electrical or optical) and the protocol (DS or SONET) of the link 17. The DS and OC protocols cited are examples only, and if other high speed protocol links are used, an appropriate interface 25 would couple the UNI to the particular link and perform the necessary interface of the UNI to the link.

AMENDED

H 05.03.00

- 11 -

One example of a chip capable of serving as the circuit 23 is ATOM4 manufactured by Digital Equipment Corporation and Toshiba Corporation. In the outgoing direction, the ATM segmentation and reassembly circuit 23 adapts information from LAN packets for ATM cell transport. The circuit 23 also maps packet address information into virtual path and circuit
5 identifiers, in order to transmit cells for particular devices or services in assigned virtual circuits over the ATM link 17.

Figure 3 is a high level functional block diagram of the ATM segmentation and reassembly circuit 23. The circuit 23 provides the functions required to implement a variety of high-performance ATM User Network Interfaces. This includes packet stream to circuit
10 selection, ATM Adaptation Layer (AAL), segmentation and reassembly (SAR), and cyclic redundancy (CRC) generation and checking. The circuit 23 contains mechanisms to support traffic shaping, varieties of ATM flow control protocols, and Operations Administration and Maintenance (OAM) flows. The ATM segmentation and reassembly circuit 23 interfaces with a variety of physical layer chips by Utopia or specific interfaces to facilitate media interface
15 connection to DS3, E3, and HSSI links.

The circuit 23 receives packets from the LAN interface 21 or the like for transmission on the ATM link 17 via a byte-wide data interface with one parity bit (TxInData) that runs at 12.5 MHz or 25 MHz. The control interface (TxIn Control) includes packet delineation and byte flow control. The flow control signal is used to hold off transmission of data during periods where no
20 buffers are available for the data to be transmitted. The receive portion of the packet interface is also byte wide for data with one parity bit (RxOut Data), and runs at the same speed as the transmit interface. There is a separate control channel (RxOut Control), which indicates receive packet delineation and error status.

The segmentation and reassembly circuit 23 connects to the actual link interface 25 through a physical link interface or port. The physical link interface consists of a byte-wide data path for both transmit (TxOut Data) and receive (RxIn Data). In all modes, the link interface is slaved from the link clock inputs. Speed matching FIFOs are used between clock boundaries. Byte transmission/reception to/from the link is controlled by either a gapped link clock or an overhead/valid indication signal on the control lines.

30 The cell memory interface consists of a 64-bit data bus (Cell Memory Data), with two additional parity bits and a 20-bit address bus (Cell Memory Address). Write enable and output enable signals are provided separately. The preferred embodiment of the ATM segmentation and

AMENDED SHEET

reassembly circuit 23 is capable of addressing up to 1 Meg word (8Mbytes) of memory. The external memory is used for storing ATM cells that are awaiting transmission or are being reassembled, records, traffic schedules and free buffer pools. The amount of memory required depends on the number of circuits being used in a particular application. Currently, 1 MB of memory is used to support 4096 circuits.

The segmentation and reassembly circuit 23 connects to the microprocessor 27 through a node processor interface or port. The node processor interface uses a 16-bit data and 7-bit address configuration. This port uses a synchronous Motorola 68K style interface, running at 12.5 or 25 MHz. All of the chip control and status registers (CSRs) can be accessed directly through this interface. The external memory 29 also is accessed indirectly through this interface.

As shown in high level form in Figure 3, the segmentation and reassembly circuit essentially comprises four processing engines, a segmentation engine 31, a cell transmit engine 33, a reassembly engine 35 and a packet relay engine 37. Figure 4 shows the elements of the segmentation and reassembly circuit 23 in somewhat more detail.

As shown in Figure 4, the segmentation engine 31 receives packets from a transmit MAC interface 41. The segmentation engine 31 segments packets into either AAL3/4 or AAL5 cells according to the virtual circuit state that is set up for each packet. At the start of a packet, the segmentation engine 31 uses a region of the header to identify the particular ATM circuit that the cells should be queued on. Each ATM virtual circuit is identified with one of several virtual path connections (VPCs) and with a specific virtual circuit connection (VCC) within the particular VPC. Up to 4095 separate circuit queues are supported. The region of the header used is programmable and depends on the packet format, e.g., the format used on the LAN 11 in the example in Figure 2.

The segmentation engine 31 supplies the cells to a memory controller and scheduler 43. The controller and scheduler 43 stores the cells resulting from the segmentation in external SRAM (memory 29) for transmission by the cell transmit engine 33. In the simplified embodiment illustrated (Figure 4), the transmit engine 33 includes a cell transmitter and scheduler 45 and transmit FIFO buffers 47 which connect to the transmit link portion 49 of the physical link interface port. Transmission of a cell segmented from a packet can occur while the rest of the packet is still being segmented and hence transmit "cut through" is supported.

The cell transmit engine 33 services up to 4095 transmit queues for CBR, VBR, ABR and unspecified bit rate (UBR) traffic. CBR and VBR traffic is

AMENDED 5:

12A

served according to precomputed traffic schedule that is

RECEIVED 5/3/00

- 13 -

stored in external SRAM, using one or more schedule tables in accord with the invention. This schedule table(s) can be used for peak-rate traffic shaping on a per circuit basis. The manner in which this table is filled out can also allow prioritization of traffic and overbooking of bandwidth. The granularity of bandwidth assigned to circuits is programmable by appropriate programming of the table(s).

A speed-matching FIFO 47 exists in the cell transmit engine 33 to account for the different clock speeds used for the core process (25 MHz) and the link process (0-25 MHz). The cells are simply output as a stream, e.g., for an STS-3c link. The byte stream output can be held off by either gapping the link clock or asserting the overhead indication signals.

The reassembly engine 35 receives cells through a receive portion 51 of the physical link interface. This engine 35 includes a FIFO 53 and a reassembly section 55. The reassembly engine 35 receives cells in byte-wide format from the link. The FIFO 53 provides speed matching between the link clock domain (0-25 MHz) and the core process clock domain (25 MHz).

Packet reassembly in section 55 begins by taking a cell from the FIFO 53 and looking up the receive circuit state that is stored in external SRAM (memory 29). The index to the receive circuit state table is based on individual lookups of the VPI field, the VCI field, and possibly, the MID field if AAL4 is being used. If the circuit lookup indicates that the cell should be accepted, the cell is written to a free cell buffer in external memory 29. If a packet reassembly is completed by the addition of this cell, the list of cells comprising the packet is moved to one of two packet lists, depending on priority bit setting for this circuit, and the circuit state is cleared. Otherwise, the circuit state is updated and written back to memory. Full AAL5 or AAL3/4 checking is done, including CRC checking, length field checking, segment type checking, and sequence number checking.

The packet relay engine 37 comprises a packet dequeuer 57. The packet relay engine 37 services two prioritized queues each of which contains a list of cell buffers that represents reassembled packets. The reassembly engine places reassembled packets into this queue and indicates to the packet relay engine when a packet is ready for servicing. Each packet contains a pointer to the circuit that it was received on. The packet relay engine 37 provides the MAC destination address in the header of the reassembled packet. The packet header and the complete packet are passed to the MAC interface 37. The RxOut MAC interface 59 uses the MAC protocol to deliver packets. This protocol delineates the packets with start and end markers and it indicates the status of the packet, e.g., good or bad CRC. In the illustrated example (Figure 2), the receive

- 14 -

MAC interface 59 supplies the reassembled packet to the LAN interface 21 for transmission over the LAN 11 to the destination device 13 or 15.

The above description of general transmit and receive operations of the ATM segmentation and reassembly circuit 23 is provided for understanding the context of the preferred implementation of the present invention. In accord with the invention, the scheduler associated with the cell transmitter 45 schedules a variety of different types of traffic on a hierarchical basis for a plurality of virtual paths within the ATM link. That scheduler implements a process similar to that of Figure 1 using one or more scheduling tables. Further discussions will concentrate on hierarchical scheduling using two implementations of the scheduling table(s).

Figure 5 depicts a simplified example of a first table embodiment, i.e. a single scheduling table, used in the hierarchical scheduler. The scheduling table includes multiple lines indexed by cell transmit time. The first field in each line contains an identifier or index for one of the VPCs. The VPC index in a line effectively assigns the cell transmit time of that line to the indexed VPC.

The next two fields in each line identify VCCs for different levels of service priority, and the succeeding two fields point to lists of additional VCCs having a different type or level of service. The VC or VCC identifiers used in this table (and other tables discussed later) may take the form of internal numbers or names arbitrarily assigned as the identifiers. In such a case, the segmentation and reassembly circuit 23 will translate those identifiers into actual ATM standard virtual circuit identifiers (VCIs) for insertion into the cells transmitted over the ATM link by the interface chip. Alternatively, the segmentation and reassembly chip 23 may utilize ATM VCIs as the internal virtual circuit identifiers.

Returning to Figure 5, the next field in each line contains an identifier for a VCC having a high priority service. The high priority service may be a VBR service, but typically the high priority service is a CBR service. The next field in the table contains an identifier for a VCC having a low priority service. Here, low priority means that the service is lower in priority than the service (if any) identified in the high priority field. In preferred implementations, the high priority service is CBR service, and the low priority service is VBR service. As discussed more below, other services, such as ABR traffic, actually have still lower priority than those identified in the 'low priority' field.

In operation, the scheduler traverses the table as it increments the cell transmit time. The order of accessing the lines of the schedule table is a-b-c-d-e-f-g-h-a... which are stored in 'next

- 15 -

serve time' in the field in each line. For each cell transmit time, the scheduler accesses one line in the table to perform a variety of scheduling functions. If the scheduler finds a CBR VCC in the high priority column and it has a cell ready to transmit, the scheduler initiates transmission of a cell over that VCC. If not, the scheduler looks to the low-priority column. If the line
5 identifies a VCC for low priority service, the scheduler checks the cell queue in memory 29 holds a cell that is ready for transmission over that VCC. If so, then the scheduler initiates transmission of a cell for that VCC. If not, the scheduler offers the cell transmit opportunity to VCCs having lower priority services.

For each cell transmit time, the scheduler also uses information from each line to schedule
10 ABR services. In the single table implementation, each line specifies a pair of pointers for ABR service. The head and tail pointers enable the scheduler to select a link list of VCCs associated with the VPC indexed in the particular line. The link list includes VCC identifiers for associated circuits carrying available bit rate service type traffic. The scheduler accesses the identified link list and appends VCCs from the link list to a work list. The scheduler may append all of the VCC
15 identifiers from the link list to the work list. Alternatively, the scheduler may append VCC identifiers only for the circuits on the link list that have a cell ready to transmit. In the single table implementation, there could be one work list for all VPCs, but preferably the scheduler uses a separate work list for the ABR traffic for each VPC. Thus, the scheduler will append the VCCs from a particular link list to the work list associated with the indexed VPC. When the scheduler
20 detects a cell transmit time for which there is no CBR or VBR service cell to transmit, the scheduler selects the VCC from the head of its work list.

Consider scheduling for several VCCs as simplified examples using the illustrated table. In cell time a, the scheduler accesses the first line in the table. Based on that line, the scheduler identifies VP1 as the VPC assigned this time slot. The scheduler checks the high priority field
25 of the first line. Different VPCs may use the same VCC identifiers, but each combination of a VPC identifier and a VCC identifier uniquely identifies one circuit. In the present example, the high priority field of the line for time a includes a VC1 identifier. This is a VCC associated with VP1 having a high priority of service. Assuming the high priority service is a CBR service and this VCC has a cell ready to transmit, the scheduler causes transmission of a cell for the VP1
30 circuit VC1 in the present time slot. The cell transmitted includes VP1 and VCI values corresponding to VP1 and VC1. During the processing for transmit time a, the scheduler will also use head and tail pointers H1, T1 to access a link list and move VCCs from that list to an ABR

- 16 -

work list associated with VP1, as discussed above.

In the illustrated example, there are four VPCs (VP1 to VP0) each having one-fourth of the link rate assigned as their respective bandwidth. To implement this allocation, each VPC has its identifier listed in one-fourth of the lines of the table. In the present example, the scheduler
5 will run through similar processes for VP2, VP3 and VP0 for time slots b, c and d respectively. In the illustrated example, each VPC identifier appears in the line corresponding to every fourth cell transmit time, e.g. VP1 appears in lines a and e. When cell transmit time e occurs, the scheduler accesses the corresponding line in the table. Based on that line, the scheduler identifies VP1 as the VPC assigned this time slot. The scheduler checks the high priority field of the first
10 line. In this example, there is no CBR service VCC listed.

The scheduler next checks the low priority field. In this example, that field identifies VC2. Here, VC2 is a different VCC associated with VP1. The low priority service preferably is a variable bit rate service. The scheduler checks to determine if this VC2 has a cell ready for transmission. If VC2 has a cell ready to transmit, the scheduler causes transmission of that cell
15 in the time slot e. The header of the transmitted cell contains a VPI corresponding to VP1 and a VCI corresponding to VC2. During the processing for transmit time e, the scheduler will also use head and tail pointers H5, T5 to access a link list and move VCCs from that list to an ABR work list associated with VP1 as in the earlier examples..

Assume now that in time e there was no cell ready to transmit over circuit VC2 for VP1.
20 In such a case, the scheduler goes to the work list for VP1 to find an ABR circuit with a cell to transmit. Assuming that the scheduler finds the VCC at the head of the work list with a cell that is ready to transmit, the scheduler causes transmission of a cell over that VCC in the time slot.

The scheduler indexes the table at the cell rate of the link. The scheduler executes the
25 hierarchical scheduling process for each line of the table. The execution order of all lines in the table is determined by the contents in the 'Next serve time' field of the table. For the example shown in Figure 5, the execution order is from time slot a to time slot b, from b to c, from c to d, from d to e, from e to f, from f to g, from g to h, from h back to a. In this fashion, the scheduler repeatedly cycles through the table to present transmit opportunities for specific circuits identified
30 in the table and in the ABR work lists.

The listing of the VCCs in the high and low priority columns of the tables allocate transmit opportunity and bandwidth to those services. Constant bit rate service is provided by

- 17 -

entries in the high priority fields of the table, and the number of entries in those fields determines the bandwidth within the VPC allocated to the particular VCC for its CBR service. In the above example, VC1 for VP1 appeared only in the high priority column. That VCC receives CBR service because the table guarantees that circuit an opportunity to transmit each time that cell
5 transmit time a occurs.

Similarly VC2 for VP1 appeared in the low priority column in a line which included no higher priority entry. That entry guarantees VC2 an opportunity to transmit each time that cell transmit time e occurs. If the VCC circuit uses the opportunity every time, it receives its maximum allowable cell transmission rate. However, at times this VCC may not have a cell that
10 is ready to transmit. If the VCC does not transmit in that slot, its transmission rate decreases, and the transmit opportunity passes to the still lower priority ABR service VCCs.

The table structure also supports a combination of CBR and VBR services for a given circuit, effectively to provide such a circuit with a variable rate service having a guaranteed minimum transmission rate. If a circuit, such as VC2 for VP3, has one or more entries in the high
15 priority column, those entries guarantee a minimum bit rate, essentially as a CBR service (see the line for time slot g). The appearance of the same VCC in the low priority (e.g. in the line for cell time c) provides an added variable rate opportunity that the circuit need not always use.

If an ABR VCC is served and this ABR needs more service, this ABR is then rescheduled to a future time slot belonging to its VPC. For this purpose, the scheduler inserts this VCC into
20 a slot, which is down the table by the amount Δt , and this slot must be associated with the same VPC. If the slot in the table indexed by Δt does not belong to the same VPC, the scheduler will then keep on searching the table until the same VPC is found down the table. ' Δt ' is computed from this ABR current service cell rate.

This form of rescheduling for ABR service is complex. Sometimes, it is not feasible due
25 to the depth of searching for the same VPC. Also, any change in bandwidth allocation requires redoing all of the table entries, which makes dynamic bandwidth allocation very difficult.

To provide a more flexible scheduling for ABR service and easier rescheduling, the second embodiment of the scheduling process uses separate tables for ABR traffic. Specifically, one static table indexes VPCs and identifies high and low priority VCCs for CBR and VBR
30 traffic. There is a separate, dynamic ABR table associated with each respective VPC. When the scheduler accesses a line of the static table for a given cell transmit time it first identifies a VPC assigned that slot for its use. The scheduler processes CBR and VBR traffic using VCC

- 18 -

identifiers from the indexed line of the static table. The scheduler also access a dynamic ABR table for the indexed VPC to retrieve a link list and append that list to a work list associated with the VPC. Moving the pointers for that list down the respective ABR table reschedules service for a specific link list of VCCs by an appropriate number of lines. It should be noted, that with
5 the second implementation, a reallocation of bandwidth between VPCs requires modification of the number of lines each VPC has in the static table. However, the ABR tables require no modification.

Figure 6 illustrates one example of a static table; and Figures 7A to 7D illustrate simplified portions of the dynamic tables, in accord with this second embodiment. Figure 8
10 illustrates process flow for an ABR work list, in the second embodiment.

As shown in Figure 6, the static scheduling table includes multiple lines indexed by cell transmit time. The scheduler traverses the static table at the cell slot time rate of the ATM link, i.e. the pointer for this table moves to the next line each time that the scheduler increments the cell transmit time. Each line of the table corresponds to one cell transmit time slot. Note that the
15 'next serve time' field does not exist any more. The scheduler traverses the static table by simply incrementing the cell time pointer.

The first field in each line contains an identifier or index for one of the VPCs. The VPC index in a line effectively assigns the cell transmit time of that line to the indexed VPC. In the illustrated example, there are four virtual path connections (VPCs), and the static table includes
20 virtual paths indexes VP1, VP2, VP3 and VP0 for those virtual paths. Each row in the table represents an opportunity for transmission of a cell for the indexed VPC. In the example, each VPC receives a transmit opportunity every fourth time slot. Thus each VPC may use up to 25% of the link bandwidth.

The next field in each line contains an identifier for a VCC having a high priority service.
25 The high priority service may be a VBR service, but again preferably relates to a CBR service. The next field in the table contains an identifier for a VCC having a low priority service. The term 'low priority' only means that the service is lower in priority than the service (if any) identified in the high priority field. Again, other services, such as ABR traffic, actually have lower priority than those identified in the 'low priority' field. The scheduler processes the high and low priority
30 entries in essentially the same manner as in the earlier embodiment.

The scheduler first presents the send opportunity to the VCC (if any) identified in the high priority field. High priority entries in the static table typically are used for CBR services. Each

- 19 -

cell time that the scheduler hits a VCC for a CBR service in the high priority field of a line, the scheduler would determine if there is a cell buffered in the memory awaiting transmission over the high priority VCC. If there is no data to send, the VCC having the CBR service still transmits a cell, albeit a cell having the VPI and VCI values assigned to the circuit in the header but having
5 no data in the payload. If the high priority field were used for some service other than CBR, e.g. VBR, then the scheduler would determine if there is a cell buffered in the memory awaiting transmission over the high priority VCC.

The high priority processing often does not cause scheduling of transmission in the respective cell time slot, typically because there is no VCC listed in the high priority field (for
10 CBR), or possibly because a VCC in that field has no cell ready for transmission on that circuit. In any such case, the scheduler passes the cell transmit opportunity to the VCC (if any) identified in the low priority field. The low priority entries typically are used for variable bit rate services. Each cell time that the scheduler hits a VCC for a VBR service in the low priority field of a line, the scheduler offers the cell transmit opportunity to that specific VCC. If the scheduler
15 determines that the identified VCC has a cell buffered in memory and ready to send at the present cell transmit time, then the scheduler will initiate transmission of the cell for that VCC. If not, then the scheduler looks to circuits having still lower priority service, such as available bit rate service, to identify a circuit and a cell to transmit.

As in the earlier embodiment, by placing identifiers for the same VCC in the high priority
20 field and the low priority field, in different lines of the table, the scheduler actually provides a variable bit rate service with a guaranteed minimum bandwidth. If the scheduler finds the VCC in the high-priority field in one line of the table, the scheduler schedules the cell transmission for that VCC during each cell transmit time that indexes that line of the static table, just as if this were a CBR service. The number of lines in the table, which include the VCC identifier in the
25 high priority field, establishes the CBR-like minimum bandwidth. Appearances of the VCC identifier in the low priority field of a number of lines provides additional bandwidth in the form of a variable bit rate portion of the service for the particular VCC.

The scheduler traverses each dynamic ABR table at the cell rate for the associated VPC. That is to say that the scheduler reads the next line of the ABR table, for a particular VPC, each
30 time that the scheduler reads the index for that VPC from a line of the static table. As such, the speed of the movement of the pointer associated with each ABR is scaled as a function of the percentage of bandwidth or cell rate allocated to the respective VPC.

- 20 -

The scheduler maintains a separate dynamic table for ABR services for each VPC that the scheduler supports. At any given time, each ABR VCC appears once and only once in a link list pointed to in the ABR table and the work list for that VPC. The bandwidth of an ABR VCC is represented by the ' Δt ', which is determined by the current cell rate of this VCC. However, the VCC can only transmit when a cell time slot is available for ABR transmission and the particular VCC identifier is at the top of the work list. Each time that an ABR VCC is presented an opportunity to transmit, it is removed from the top of the work list by the scheduler and moved down the ABR table to a new location, if this time this VCC sends a data cell, not a null cell. The new distance down the table to the new location determines the time delay until the next retrieval and placement of the VCC on the work list and thus the next opportunity for this ABR VCC to send a cell.

In the example under discussion, there are four VPCs, therefore the scheduler maintains four dynamic ABR tables, portions of which appear in Figures 7A to 7D. Each ABR table has its own current pointer. The number of lines is the number of cell times assigned to the associated VPC in the static table. For example, in Figure 6, the static table shows assignment of times a and e to VP1. The dynamic ABR table for VP1 (Figure 7A) therefore has lines corresponding to a and e.

Each line of an ABR table includes head and tail pointers H, T. The pointers identify a link list of VCCs associated with the particular VPC. Figure 8 shows the flow of the link list and work list processing. In the illustrated example, at time a, the scheduler reads H1, T1 from the ABR table for VP1. The scheduler uses those pointers to retrieve a list of ABR VCCs whose service times are due at this time slot. In the example, the retrieved list contains identifiers for VC4, VC5 and VC7 associated with VP1. The scheduler appends those VCC identifiers to the work list. In the example illustrated, VC6 and VC8 are presently in the work list for VP1 and VP6 is at the top (head) of the work list. The scheduler appends the new link list to the bottom of that work list.

When a transmit opportunity for VP1 arises, if there is no higher priority service ready to transmit, the scheduler picks up the VCC from the head (top) of the work list for VP1 to serve. In the example illustrated, VC6 is picked to serve. Ideally, the scheduler could go through the work list and find the first VCC that has a cell ready to send. But in reality, this design is not feasible for real-time processing. In the second embodiment presented here, only the VCC at the top or head of the work list is examined. If the VCC picked has no cell to send, a null cell is sent

- 21 -

instead, and this VCC will not be rescheduled to the dynamic table. Assume for this example that VC6 has a cell ready to send. The scheduler initiates transmission for the cell for that circuit and reschedules service for VC6. In the future time slot, VC6 is either appended to the tail or added at the head of the linked list in the future time slot. The precise placement in the dynamic table
5 depends on the applicable rescheduling algorithm.

More specifically, in this second embodiment, moving the pointers for an ABR link list down the ABR table by some distance ΔD reschedules service for the VCCs on that link list. As noted earlier, the scheduler traverses the ABR table for a particular VPC at a fraction of the cell rate corresponding to the fraction of the link rate allocated to that VPC. The rescheduling
10 therefore also must be scaled to the appropriate percentage of bandwidth or link rate allocated to the VPC. In the preferred process, that distance ΔD equals the percentage bandwidth for the associated VPC multiplied by the reschedule time (ΔT) for this ABR VCC. The reschedule time (ΔT) equals the link rate divided by the allowed cell rate (ACR) for the ABR service to the particular VCC. The ACR value varies dynamically depending on traffic.

15 The second embodiment greatly simplifies modifications of the schedule table. In an actual implementation, the scheduler stores two copies of the scheduling table (Figure 5) or two copies of the static table (Figure 6). One copy of the table is active, and the other is not. To reprogram the service hierarchy, the microprocessor modifies the inactive copy of the table and transfers and changes the active/inactive status of the two tables. In the second embodiment,
20 Figures 6 and 7, the dynamic tables need not be changed. It may be helpful to consider some examples on this point.

In the second implementation, tear-down or set-up of a real-time connection requires modification of the static schedule table. One control bit in a control and status register indicates which of the two static tables currently is in use. The microprocessor negotiates service with the
25 appropriate data device 13 or 15 to define the circuit to be set up or torn down, and modifies the inactive static table to reflect this change. After completion of modification of this table, the microprocessor flips the control bit to activate usage of the new static table and to deactivate the other table. When the current time pointer reaches the end of the old table and wraps around to the cell time for the top of the table, the base of the new static table is copied to the current time
30 pointer, and the scheduler begins using the new static table.

To set up or tear down a non-real time connection initiates a treatment of the circuit by modifying the rescheduling process in the appropriate dynamic, ABR table. For example, to set

- 22 -

up such a connection, the microprocessor modifies a status register associated with the circuit. When the circuit first has traffic to send, the pointer pointing to the circuit is added to the work list for the appropriate VPC. To tear down a connection, the status register for the circuit is changed to inactive. After the next service opportunity is offered to the circuit, the scheduler will
5 not reschedule service for that circuit in the ABR table.

As discussed above, the number of appearances of a VPC identifier in the VP index of the static table determines the bandwidth allocation to the VPC. To modify the bandwidth for a VP, the microprocessor changes the inactive copy of the static table to include the VPC in the appropriate number of lines for the new level of bandwidth and then activates the modified copy
10 of the table in the manner outlined above.

If the scheduler completes the static table, dynamic table and work list processing without finding a circuit with a cell to send, the transmit opportunity passes down to one or more lower level service circuits. In the preferred implementation, the segmentation and reassembly circuit supports a number of other types of flow control, such as quantum flow control. For simplicity
15 of discussion, however, assume that there is one lower level service, quantum flow control. Quantum flow control (QFC) processing maintains a 'credit' for each VCC subscribing to the service. For each VPC, the scheduler maintains a separate queue of VCCs for QFC based services. The scheduler identifies the VPC from the VP index in the static table and performs its processing for CBR, VBR and ABR, as discussed earlier. If the VPC assigned to a particular cell
20 time has no cell ready to send for any VCC having those services, the scheduler looks to a queue of VCCs for QFC, associated with the currently identified VP, to fetch the VCC at the head of the queue.

When a QFC opportunity to transmit arises with respect to one VPC, the scheduler initiates transmission of the cell for that VCC. After transmission, if the VCC has more cells to
25 send and it has more credits, it is appended to the tail of the QFC list for associated VPCs. Otherwise, it is simply removed from the QFC list. Next time, when it has traffic coming in and it also has credit updated, this VCC is appended to the tail of the QFC list. In this manner, all VCCs in a QFC list have both credits and a cell to send. When a transmission opportunity comes, all VCCs are ready to send a cell.

30 To ensure detailed understanding, it may be helpful to go through a specific process flow, with emphasis on the processing of the second embodiment of the scheduling tables. Figure 9 is a flow chart illustrating this scheduling process.

The scheduler indexes the cell transmit time (S111) and accesses the line in the static table corresponding to the indexed cell time. From the first field in the line of the static table for that time slot, the scheduler identifies the VPC assigned that time slot. The scheduler next checks the high priority field of that line (S112) and determines if there is a VCC listed (S113). If there is
5 a VCC listed in that field and this VCC has a cell ready to send, the identified VCC has a CBR service allocated to use of that time slot. Under these circumstances, the scheduler sets a bit indicating CBR ready and assigns this cell transmit slot to the identified CBR VCC (S114-1).

If the high priority field in the indexed line does not include a VCC for a CBR service or the listed CBR VCC has no cell to send, however, the scheduler next accesses the low priority
10 field in the table (S114-2). The low priority field is used to identify VCCs for variable bit rate services. In any given line of the table, there may or may not be a low priority entry. If there is an entry in this field, the scheduler checks the circuit identified to determine if that circuit has a cell to send (S115). Under these circumstances, the scheduler sets a bit indicating VBR ready and assigns this cell transmit slot to the identified VBR VCC (S116). If there is no VCC listed in the
15 low priority field or if a VCC is listed but has no cell to send, then the scheduler will not set the ready bit for ABR transmission.

The scheduler next reads the head and tail pointers from the indexed line of the ABR table for that VPC (S117). The head and tail pointers point to a link list of VCCs having ABR service. Using the pointers, the scheduler retrieves the link list and may move one or more of the VCCs
20 contained in the particular link list to the work list for the currently identified VPC (S118). There are a number of possible algorithms that the scheduler may use to develop the work list. For example, a given link list may have some special priority requiring placement in the middle of the work list, at the top of the work list or at some other specified location on the work list. The scheduler may check to determine which circuits on the link list have a cell ready to transmit and
25 transfer the VCC identifiers to the work list only for those circuits (if any) that have a cell to transmit. For simplicity of discussion here, assume that the scheduler appends all of each newly accessed link list of VCCs to the bottom of the work list as in the example of Figure 8.

The scheduler now checks the ready status (S119). If either the CBR ready bit or the VBR ready bit is set, the scheduler skips forward (to step S130), without performing any processing
30 on the ABR work list for the indexed VPC. However, if neither the CBR ready bit nor the VBR ready bit is set, the scheduler looks to the VCC at the head of the work list (STEP 120).

In the current example, the scheduler checks to determine if the first VCC in this list has

AMENDED SHEET

- 24 -

a cell ready to transmit. If the first VCC in the work list has a cell to transmit, the scheduler sets a bit indicating ABR ready, assigns this cell transmit slot to the identified ABR VCC and removes the VCC from the work list. If the first VCC identified on the work list does not have a cell ready to transmit, the scheduler removes the VCC from the work list, but it does not set the ABR-ready bit (S122) and S122 is skipped. This process continues until the scheduler identifies an ABR VCC from the work list with a cell to transmit or until the scheduler completes traversal of the ABR work list for the indexed VPC without identifying a VCC that is ready to transmit (S121). If the scheduler finds a VCC having ABR service on this list with a cell to transmit, the scheduler sets the ABR-ready bit and assigns the slot to that VCC (S122). If the scheduler finds no VCC on the work list with a cell to send, the scheduler skips step S122.

The scheduler reaches step S130 either upon recognition of a ready status set for CBR or VBR (from S119) or upon completion of the ABR work list processing for the indexed VPC (from S121 or S122). At this point, the scheduler again checks ready status. If a ready bit is set for CBR, VBR or ABR, the scheduler jumps forward to its transmit process in step S200 and initiates transmission of the appropriate cell.

However, if the scheduler completed the work list for the ABR traffic for the currently indexed VPC without identifying a VCC with a cell to send, there will be no ready bit set when the scheduler checks status in step S130. The scheduler next considers QFC traffic for the indexed VPC (S131, S132). The scheduler checks if the QFC list for the indexed VPC is empty (S131, S132). If the QFC list is not empty, the scheduler initiates transmission of a cell for the VCC at the head of the list, and sets a bit indicating QFC ready (S133). The scheduler jumps forward to its transmit process in step S200 and initiates transmission of the appropriate cell.

However, if at step S132 the scheduler has found no VCC on the QFC queue, the scheduler skips step S133.

The scheduler may go through tables and/or work lists for other types of traffic associated with the indexed VPC. For purposes of the present discussion, it is assumed that the QFC traffic is the lowest level service in the hierarchy associated with that VPC.

The virtual path identified as VP0 has special transmit privileges. It can transmit CBR, VBR and ABR traffic as scheduled by the tables and work list, in the same manner as the other VPCs. However, VP0 may also transmit ABR or QFC traffic in cell transmit time slots designated for other VPCs, any time that the VPC identified in the VP index field of the static schedule table does not have a cell ready to transmit. In the illustrated example, it is assumed for

discussion that the indexed VPC was for a connection other than VP0. After going through all of the processes for the indexed VPC (CBR, ABR and QFC in the present example), the scheduler offers the cell transmit opportunity to ABR traffic associated with VP0. The scheduler checks the ABR work list (S134, S135) for VP0 in the same manner as for other VPCs. If the VCC at the head of the ABR work list for VP0 has a cell ready to transmit, the scheduler sets the ABR-ready bit during processing for VP0, (STEP 136) the scheduler advances its processing to the transmission operation (S200) and sends the cell over the identified VCC for VP0.

If there is no ABR service VCC for VP0 with a cell to send, then the scheduler goes to the QFC queue for VP0 (at S137 and S138). If the QFC queue for VP0 is not empty, the scheduler sets the bit indicating QFC ready (S139), and proceeds to the transmission operations for the VCC at the top of the QFC list (S200).

If the scheduler finds an empty QFC queue for VP0, the scheduler may go to a work list or queue for another type of traffic, such as flow master or unspecified bit rate (UBR) (S140, S141). If the scheduler finds a UBR VCC ready for cell transmission in the appropriate list for VP0, the scheduler updates that work list. If the VCC was selected from an ABR list, the scheduler removes the pointer pointing to that VCC, that is the VCC itself, from the top of the work list and moves it down the ABR table for the particular VPC, to reschedule service for that VCC.

In the present example, there may be UBR traffic for VP0 but no other traffic types, therefore if there is no UBR traffic scheduled for VP0, none of the circuits offered this cell transmit opportunity have a cell to send. Accordingly, the scheduler initiates transmission of a null cell (S143).

In step 200, if the scheduler set any of the ready bits, the scheduler uses the appropriate identification for the VCC that caused the setting of the particular ready bit and initiates transmission of the cell for that VCC. The cell transmitted includes a VPI corresponding to the indexed VPC and a VCI corresponding to the identified VCC, to logically place the transmitted cell in the virtual path and in the virtual circuit within that path. Thus, if the scheduler set the CBR-ready bit, the scheduler initiates transmission of the cell for the VCC identified as the CBR service VCC for this cell transmit time in the high priority field of the indexed line of the static table for the indexed VPC. If the scheduler set the VBR-ready bit, the scheduler initiates transmission of the cell for the VCC identified as the ABR service VCC for this cell transmit time in the low priority field of the indexed line of the static table for the indexed VPC. If the

AMENDED SHEET

- 26 -

scheduler set the ABR-ready bit during processing of the ABR work list for the indexed VPC, then the scheduler initiates transmission of the cell for the VCC from that list. If the scheduler set the QFC-ready bit during processing of the QFC queue for the indexed VPC, then the scheduler initiates transmission of the cell for the VCC from that list. If the scheduler set the ABR-ready bit during processing of the ABR work list for VP0, then the scheduler initiates transmission of the cell for the VCC from that list. If the scheduler set the QFC-ready bit during processing of the QFC queue for VP0, then the scheduler initiates transmission of the cell for the VCC from that list. If the scheduler set the UBR-ready bit during UBR processing for VP0, then the scheduler initiates transmission of the cell for the VCC having the UBR service.

10 If the cell transmission was for a VCC from one of the ABR or QFC lists, the scheduler updates that list. If the VCC was selected from an ABR list, the scheduler removes the link list containing that VCC from the particular work list and moves the head and tail pointers down the ABR table for the particular VPC, to reschedule service for that VCC. Also, the scheduler similarly removes pointers and reschedules service for any link lists passed over in the work list processing because there were no cells ready to transmit for the identified circuits. The distance
15 down the ABR table that the scheduler moves the pointers for each link list depends on the scheduling algorithm, traffic and/or the level of ABR service provided to the particular VCC. If the VCC was selected from a QFC list, then the scheduler moves the VCC down that queue and deducts one transmission from the credit assigned to the VCC. Periodically, the scheduler
20 refreshes QFC credit for each VCC.

Next, the scheduler increments the cell transmit time (S111) and begins the scheduling process again. In particular, the scheduler will read the next line in the static table and begin processing for the VPC identified in that next line. The current pointer for the ABR table for that VPC also increments to the next line.

25 From the above discussion, it should be apparent that additional work lists can be provided to support other types of traffic, either associated with the indexed VPC, associated with VP0 or independent from any identified VPC. If the additional queue or work list process is added after the QFC processing for the indexed VPC but before VP0 processing, the list would relate to service associated with the one VPC. If added after the VP0 processing, the service may
30 be one associated with VP0 or independent of VP. However, such a service is at the bottom end of the hierarchical scheduling process. VCCs identified in such a bottom-end process would receive service only after all CBR and VBR service checks for the indexed VPC and after offering

- 27 -

the transmit opportunity to VCCs on the ABR work lists and QFC work lists for both the indexed VPC and VP0.

To recap, it may be helpful to consider the following summary of operation of a working embodiment. Each CBR or VBR VCC has a 'ready bit' to indicate if the circuit has data cells ready. This 'ready bit' could be kept on external memory (e.g. memory 29), but preferably it is kept on the scheduler chip. When cells of either CBR or VBR circuits are ready to send, the scheduler sets the 'ready bit' for that VCC to one. Every time a cell of a CBR or VBR VCC is transmitted, this VCC's 'ready bit' is cleared to zero.

Each VP has a work list for its ABR VCCs. Each ABR VCC appears once and only once in either the work list or one of the linked lists identified in the ABR schedule table, if this VCC has a cell to send. When cells for an ABR VCC are ready to send, if this VCC is not in any list, neither the work list nor a linked list, this VCC is appended to the tail of its VPC's work list. Otherwise, nothing happens. When a cell of an ABR VC is transmitted, if this VCC still has cells to send, its service then is rescheduled to the future time. To do so, the VCC is put into a linked list whose head and tail pointer are in the future time slot in the ABR table. If this VCC has nothing more to send, it is not put into any list.

At any cell time slot, the scheduler always checks the high priority VCC column in the static table first. If there is a VCC scheduled in the high priority column and this VCC has a cell to send, indicated by 'ready bit' for this circuit, the cell transmission time is given to this high priority VCC. Otherwise, the transmission opportunity is passed to the low priority column. For example, if the cell time pointer is pointing to cell time A, the static table (Figure 6) lists VC1 in the high priority column and VC4 in the low priority column. If the 'ready bit' for VC1 is set, meaning that there is a cell ready for transmission, VC1 is served. If 'ready bit' for VC1 is not set, the transmission opportunity passes to VC4 in the same row. If the cell time pointer is pointing to cell time B, there is no VCC in the high priority column and the transmission opportunity passes to the VC3 listed in the low priority column. At any cell time slot, if there is no VCC in the high priority column, or the VCC in the high priority column has no data to send, the scheduler checks the VCC in the low priority column. If there is a VCC in the low priority column and the 'ready bit' for that VCC is set, a cell is transmitted for that VCC. Otherwise, an ABR linked list from the ABR table for this VPC is checked. For example, if the cell time pointer is pointing to cell time B and VC3 has its 'ready bit' set, a cell for VC3 is sent. Otherwise,

- 28 -

the ABR schedule table for VP2 is checked. At any cell time slot, the linked list in the current served VP's ABR table pointed by its current pointer is appended to this VP's work list. If there is no service request from either the high priority VCC or the low priority VCC from the static table, the VCC in the head of this VP's work list is served. The current pointer of the ABR table
5 increments.

Consider an example in which the cell time pointer is pointing to cell time B, the current served VP is VP2. In VP2's ABR table, the current pointer p2 is pointing to the linked list with head and tail H6 and T6, respectively. This linked list is appended to the work list of VP2. There is no VCC in the high priority VCC column and the 'ready bit' for VC3 of the low priority VCC
10 column is not set. So the VCC at the head of the work list of VP2 is then served in the current slot time. If this VCC has more cells to send, it is rescheduled to the future time, i.e. another slot in the ABR table for VP2. Depending on the algorithm used, the rescheduled VCC could be inserted to the head of the linked list in the future time slot or the tail of that list. The current pointer p2 proceeds to the next line containing the linked list with head and tail H9 and T9,
15 respectively. When the current pointer p2 reaches the end of this ABR table, it goes back to the beginning of the table.

It will be readily seen by one of ordinary skill in the art that the present invention fulfills all of the objects set forth above. After reading the foregoing specification, one of ordinary skill will be able to effect various changes, substitutions of equivalents and various other aspects of
20 the invention as broadly disclosed herein. It is therefore intended that the protection granted hereon be limited only by the definition contained in the appended claims and equivalents thereof.

- 29 -

CLAIMS

1. A hierarchical method of scheduling asynchronous transfer mode (ATM) cell traffic for transmission over a link, in a plurality of cell transmit times, the method comprising the following steps in each respective cell transmit time:

5 (a) accessing a line of a multi-line static scheduling table to identify one virtual path connection that is assigned the respective cell transmit time, from among a plurality of virtual path connections that may utilize bandwidth on the link;

(b) determining from the accessed line if there is a virtual circuit connection from among a plurality of virtual circuit connections associated with the one virtual path connection having
10 a high priority service that is assigned the respective cell transmit time and if the virtual circuit connection has a cell ready to send, and if so, presenting an opportunity to transmit a cell for that high priority service virtual circuit connection over the link in the respective cell transmit time;

(c) indexing a line of a multi-line dynamic scheduling table associated with one virtual path connection;

15 (d) using data from the indexed line in the dynamic scheduling table to identify a list of available bit rate service virtual circuit connections associated with the respective cell transmit time;

(e) adding the identities of available bit rate service virtual circuit connections from the identified list to a work list associated with the one virtual path connection; and

20 (f) if no cell was transmitted for a high priority service virtual circuit connection in the respective cell transmit time, accessing the work list to transmit a cell for a specifically selected available bit rate service virtual circuit connection over the link in the respective cell transmit time.

2. The method as recited in claim 1, further comprising moving the data from the indexed
25 line to another line of the dynamic scheduling table, to reschedule service for the specifically selected available bit rate service virtual circuit connection.

3. The method as recited in claim 1, wherein step (c) comprises accessing a link list associated with the one virtual path connection.

- 30 -

4. The method as recited in claim 3, wherein the step of accessing a link list comprises: obtaining a pointer from a field of the indexed line of the dynamic scheduling table and using the pointer to retrieve the link list from a memory containing a plurality of link lists.

5. The method as recited in claim 4, further comprising the step of moving the pointer
5 from the indexed line of the dynamic scheduling table to a new position in the dynamic scheduling table, after transmitting a cell for the specifically selected available bit rate service virtual circuit connection so as to reschedule service relating to the specifically selected available bit rate service virtual circuit connection.

6. The method as recited in claim 1, wherein the determining step comprises reading a
10 high priority identifier field in the accessed line of the static scheduling table to determine if the field contains an identifier for a virtual circuit connection.

7. The method as recited in claim 6, wherein the step of presenting an opportunity to transmit comprises transmitting a cell for the high priority service virtual circuit connection over the link in the respective cell transmit time so as to provide a constant bit rate service.

15 8. The method as recited in claim 1, wherein the step of presenting an opportunity to transmit comprises:

determining if a cell is ready for transmission in the respective cell transmit time for the high priority service virtual circuit connection; and

only if there is a cell ready for the high priority service virtual circuit connection,
20 transmitting the cell for the high priority service virtual circuit connection over the link in the respective cell transmit time so as to provide a variable bit rate service.

9. The method as recited in claim 1, wherein step (f) comprises selecting the virtual circuit connection from a head of the work list as the specifically selected available bit rate service virtual circuit connection.

25 10. A device for scheduling asynchronous transfer mode (ATM) cell traffic for

- 31 -

transmission over a link, in a plurality of cell transmit times, the device comprising:

a static scheduling table stored in memory, said static scheduling table comprising a plurality of lines, each of which indexes one of a plurality of virtual path connections from among a plurality of virtual path connections that may utilize bandwidth on the link and includes two
5 fields that may contain identifiers of a plurality of virtual circuit connections associated with the then indexed virtual path connection;

means for sequentially accessing the lines of the static scheduling table to identify one virtual path connection that is assigned to each respective cell transmit time;

means responsive to data from the field of a line of the static scheduling table for
10 determining if there is a virtual circuit connection associated with the one virtual path connection having a high priority service that is assigned the respective cell transmit time and if this virtual circuit connection has a cell ready to send, and if so, presenting an opportunity to transmit a cell for that high priority service virtual circuit connection over the link in the respective cell transmit time;

15 a plurality of link lists stored in memory;

a dynamic scheduling table indexing pointers to the link lists;

means responsive to the dynamic scheduling table for accessing a link list from the memory, the accessed link list identifying at least one available bit rate service virtual circuit connection associated with the respective cell transmit time from among a plurality of available
20 bit rate service virtual circuit connections associated with the one virtual path connection; and

means for selecting one identified available bit rate service virtual circuit connection for transmission of a cell over the link in the respective cell transmit time, if no cell was transmitted for a high priority service virtual circuit connection in the respective cell transmit time.

11. The device as recited in claim 10, comprising a scheduler in an ATM segmentation
25 and reassembly circuit.

12. The device as recited in claim 11, wherein the ATM segmentation and reassembly circuit is implemented in an ATM user network interface.

13. The device as recited in claim 10, further comprising a work list stored in memory,

- 32 -

wherein the accessing means appends an identification of at least one virtual circuit connection from the accessed link list to the work list, and the selecting means selects one virtual circuit connection from the work list.

14. The device as recited in claim 13, wherein the work list stored in memory comprises
5 a plurality of separate work lists, each separate work list being associated with one of the virtual path connections that may utilize bandwidth on the link.

15. The device as recited in claim 10, further comprising means for moving a pointer for the selected virtual circuit connection down the dynamic scheduling table.

16. A method of scheduling asynchronous transfer mode (ATM) cell traffic for
10 transmission over a link, the method comprising:

(a) indexing a line of a static scheduling table for a respective one of a plurality of cell transmit times;

(b) reading data from the indexed line of the static scheduling table to identify one virtual path connection from among a plurality of virtual path connections that may utilize bandwidth
15 on the link;

(c) reading data from the indexed line of the static scheduling table to determine if there is a high priority virtual circuit connection that is assigned the respective cell transmit time from among a plurality of high priority virtual circuit connections associated with the one virtual path connection, and if there is an assigned high priority virtual circuit connection and the connection
20 has a cell ready to send, transmitting a cell for the assigned high priority virtual circuit connection over the link in the respective cell transmit time;

(d) if there is no assigned high-priority virtual circuit connection or the assigned high-priority virtual circuit connection has no cell to send for the respective cell transmit time, reading data from the indexed line from the static table to determine if there is a low-priority virtual
25 circuit connection that is assigned the respective cell transmit time from among a plurality of low priority virtual circuit connections associated with the one virtual path connection, and providing an opportunity to transmit a cell for the assigned low priority virtual circuit connection over the link in the cell transmit time;

- 33 -

(e) indexing a line of a dynamic scheduling table, associated with the one virtual path connection, for the respective cell transmit time.

(f) using data from the indexed line of the dynamic scheduling table to identify at least one available bit rate service virtual circuit connection from among a plurality of available bit rate service virtual circuit connections associated with the one virtual path connection;

(g) adding the identity of the at least one available bit rate service virtual circuit connection to a work list; and

(h) if no cell was transmitted for either a high priority virtual circuit connection or a low priority virtual circuit connection in the respective cell transmit time, accessing the work list to identify an available bit rate service virtual circuit connection, and transmitting the cell for the identified available bit rate service virtual circuit connection over the link in the respective cell transmit time.

17. The method as recited in claim 16, further comprising the step of rescheduling a cell transmit opportunity for any available bit rate service virtual circuit connection that transmits a cell for future addition to the work list in accord with a predetermined scheduling algorithm.

18. The method as recited in claim 17, wherein the step of rescheduling comprises moving the data from the indexed line of the dynamic scheduling table down a predetermined distance to another line of the dynamic scheduling table for indexing during another of the cell transmit times.

19. The method as recited in claim 16, wherein the data from the indexed line of the dynamic scheduling table comprises a pointer, and step (f) comprises using the pointer to identify a link list from among a plurality of link lists contained in a memory, the identified link list identifying the at least one available bit rate service virtual circuit connection.

20. An asynchronous transfer mode (ATM) processing device, comprising:
an ATM transmitter for transmitting ATM cells over a communication link;
a memory;
a cell transmission scheduler for scheduling the transmission of the ATM cells over the

- 34 -

link;

a static scheduling table, stored in the memory, for use by the cell transmission scheduler, the static scheduling table comprising a plurality of lines, each line of the static scheduling table comprising an index identifying one of a plurality of virtual path connections that may utilize the link during a respective cell transmit time, and a field for an identifier of a virtual circuit connection that may have a high priority service;

a plurality of link lists stored in memory, each link list identifying one or more virtual circuit connections; and

a plurality of dynamic scheduling tables maintained in memory, each of which is associated with one of the virtual path connections, each dynamic scheduling table comprising a plurality of lines indexed by cell times assigned in the static scheduling table to the associated virtual path connection, each line in each dynamic scheduling table comprising at least one field for a pointer to a link list.

21. The device as recited in claim 20 wherein each line in the static scheduling table further comprises another field for an identifier of a virtual circuit connection that may have a low priority service.

22. The device as recited in claim 21, wherein the scheduler comprises means for processing the fields in the static table and controlling the ATM transmitter to provide constant bit rate service as the high priority service to identified virtual circuit connections and to provide variable bit rate service as the low priority service to identified virtual circuit connections.

23. The device as recited in claim 22, wherein the scheduler further comprises means for using the pointers to provide available bit rate services to virtual circuit connections identified in the link lists.

24. The device as recited in claim 20, further comprising:

a media access control interface for receiving data relating to a plurality of data communication services from one or more data devices; and

a segmentation processor for segmenting data received via the media access control interface and for adapting the segmented data into ATM cells.

M 18 03 99

1/9

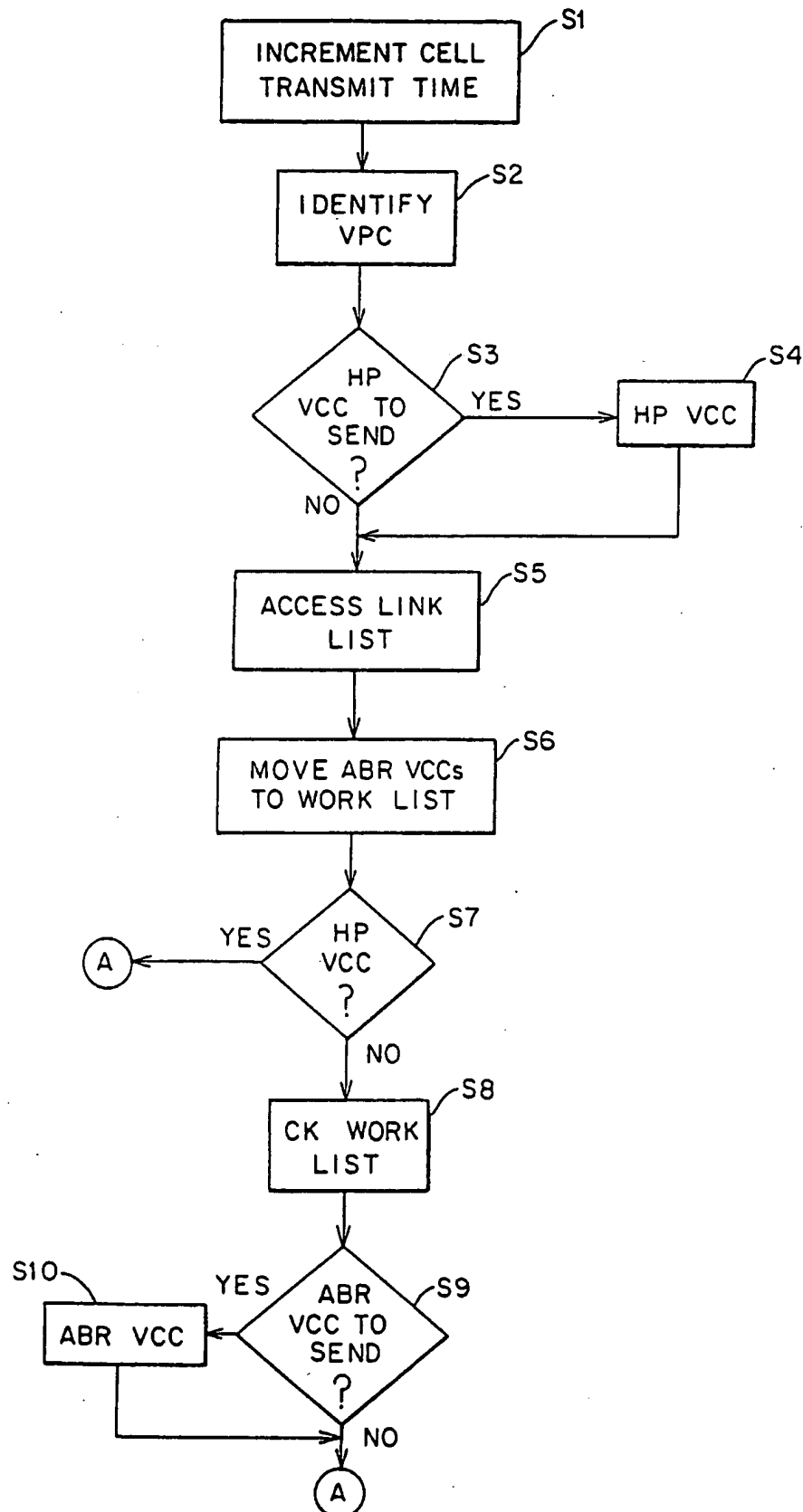
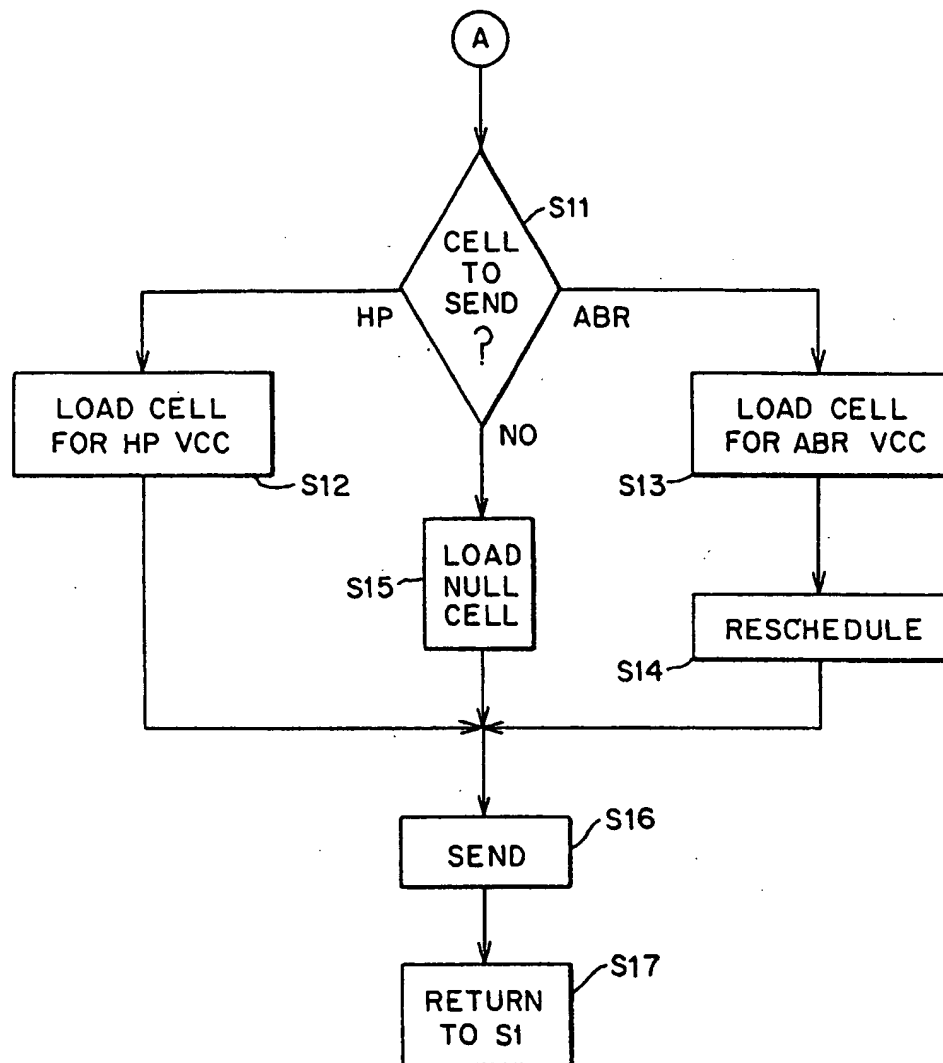


FIG. 1A

AMENDED SHEET

2/9

**FIG. 1B**

3/9

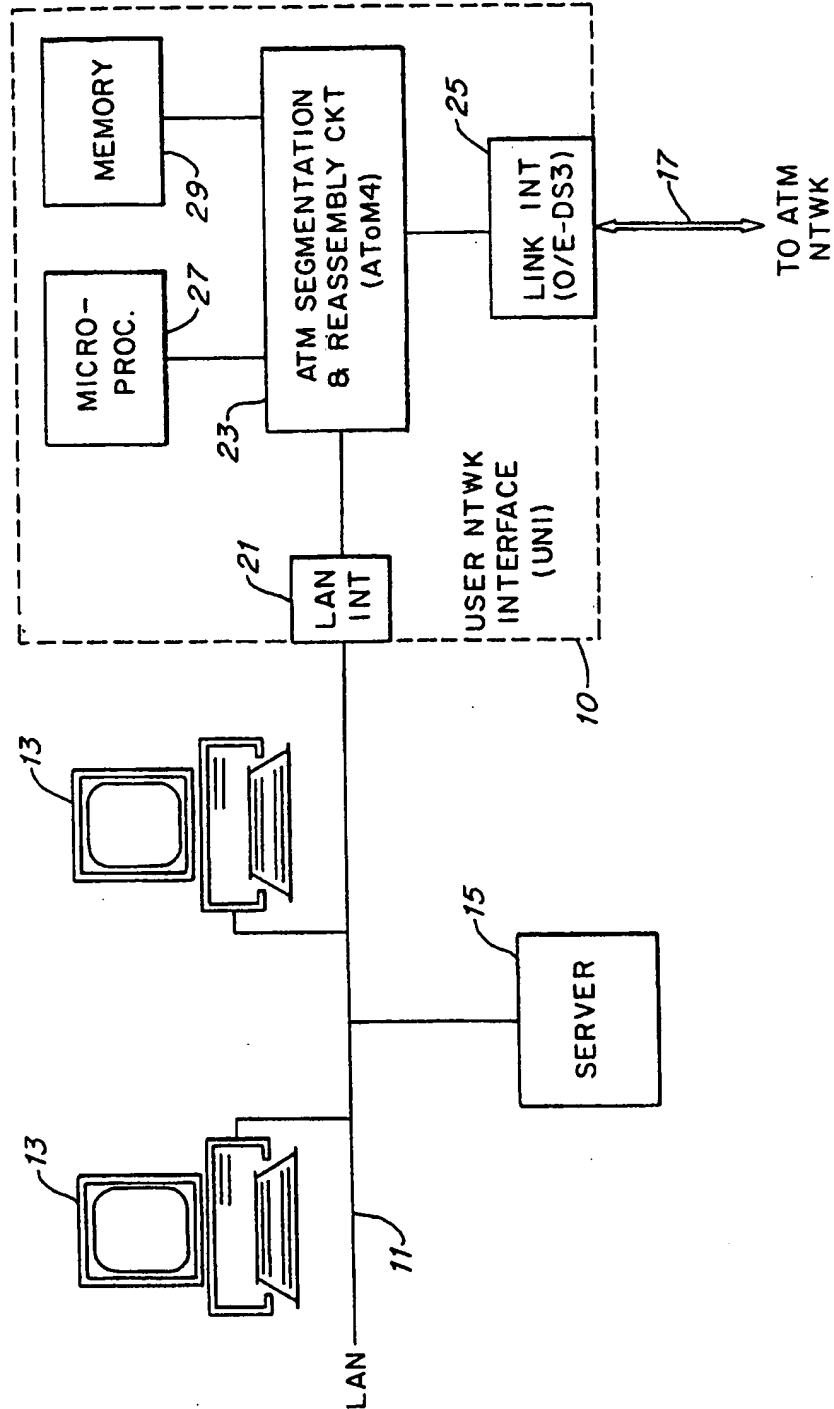


FIG. 2

4/9

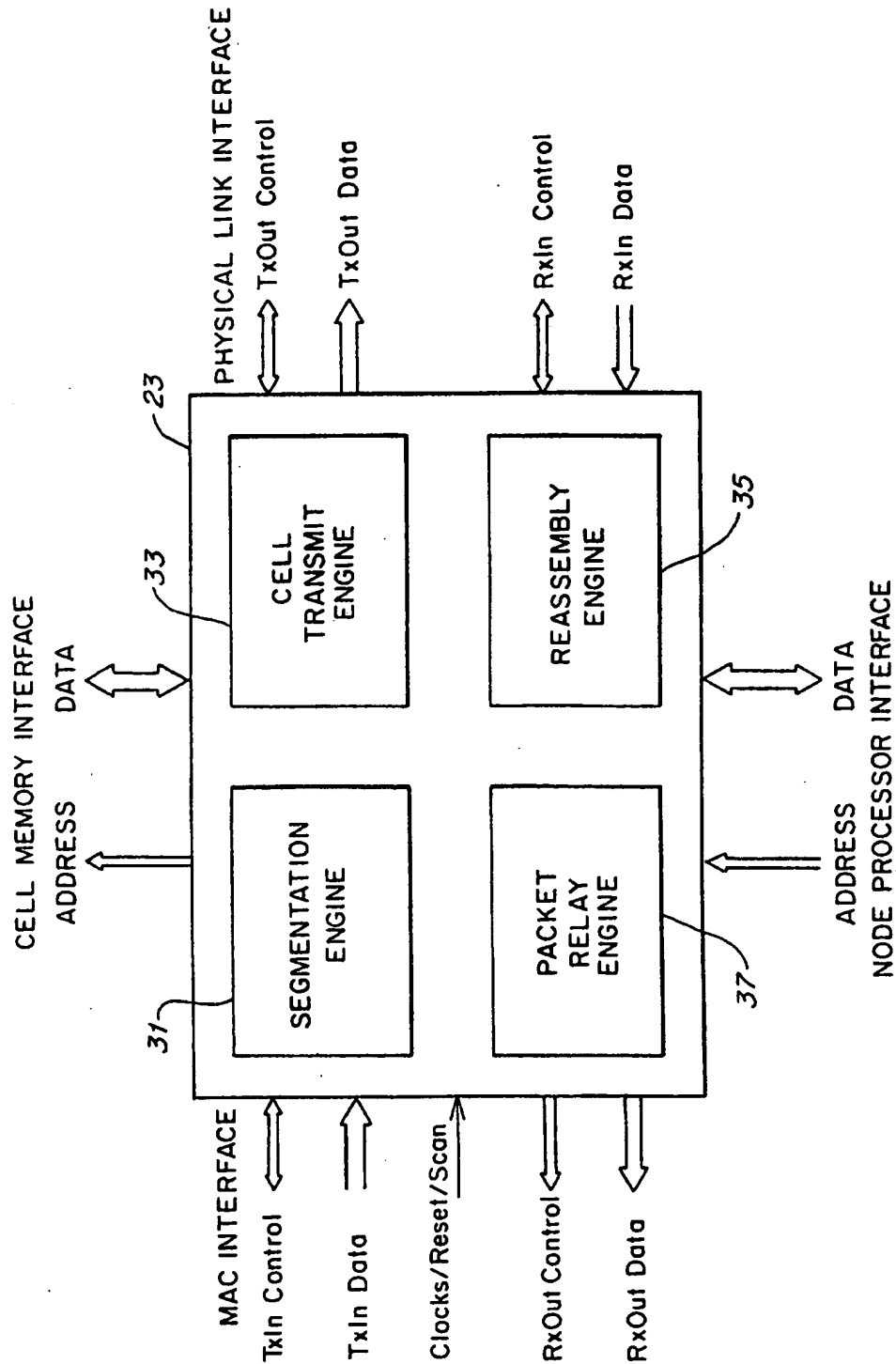


FIG. 3

SUBSTITUTE SHEET (RULE 26)

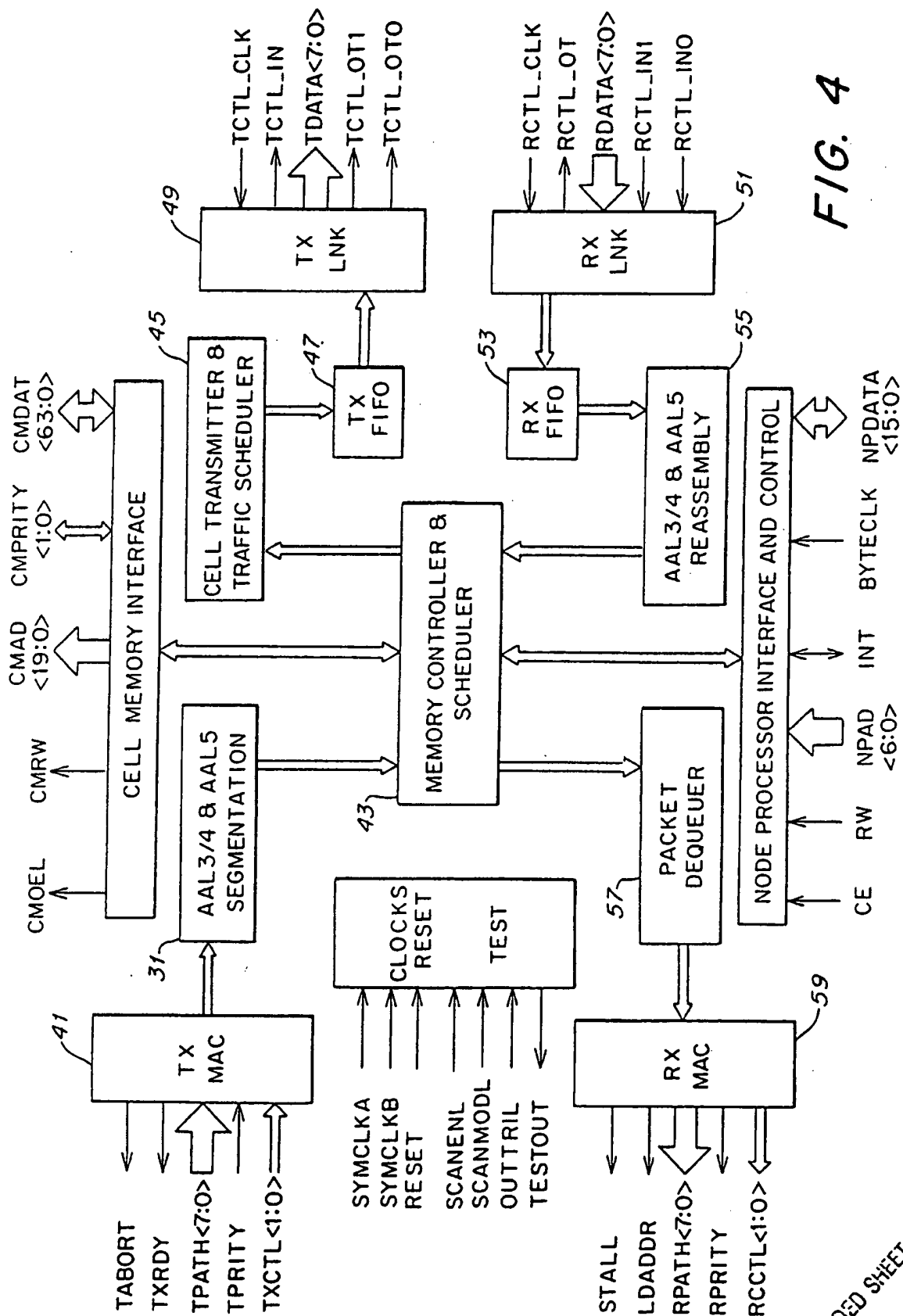


FIG. 4

AMENDED SHEET

6/9

CELL TIME	VPC	HIGH PRIORITY	LOW PRIORITY	ABR POINTERS		NEXT SERVE TIME
a	VP1	VC1		H1	T1	b
e	VP1		VC2	H5	T5	f
b	VP2		VC3	H2	T2	c
f	VP2		VC1	H6	T6	g
c	VP3		VC2	H3	T3	d
g	VP3	VC2		H7	T7	h
d	VPO		VC1	H4	T4	e
h	VPO	VC2		H8	T8	a

SERVICE ORDER a-b-c-d-e-f-g-h-a

FIG. 5

STATIC TABLE

CELL TIME	VP INDEX	HIGH PRIORITY	LOW PRIORITY
a	VP1	VC1	VC4
b	VP2		VC3
c	VP3		VC2
d	VPO		VC1
e	VP1		VC2
f	VP2		VC1
g	VP3	VC2	
h	VPO	VC2	

CELL TIME POINTER→

FIG. 6

7/9

ABR TABLE-VP1

CURRENT
POINTER P1→

ABR HEAD POINTER	ABR TAIL POINTER
H1	T1
H5	T5
•	•
•	•

FIG. 7A

ABR TABLE-VP2

CURRENT
POINTER P2→

ABR HEAD POINTER	ABR TAIL POINTER
H2	T2
H6	T6
H9	T9
•	•

FIG. 7B

ABR TABLE-VP3

CURRENT
POINTER P3→

ABR HEAD POINTER	ABR TAIL POINTER
H3	T3
H7	T7
•	•
•	•

FIG. 7C

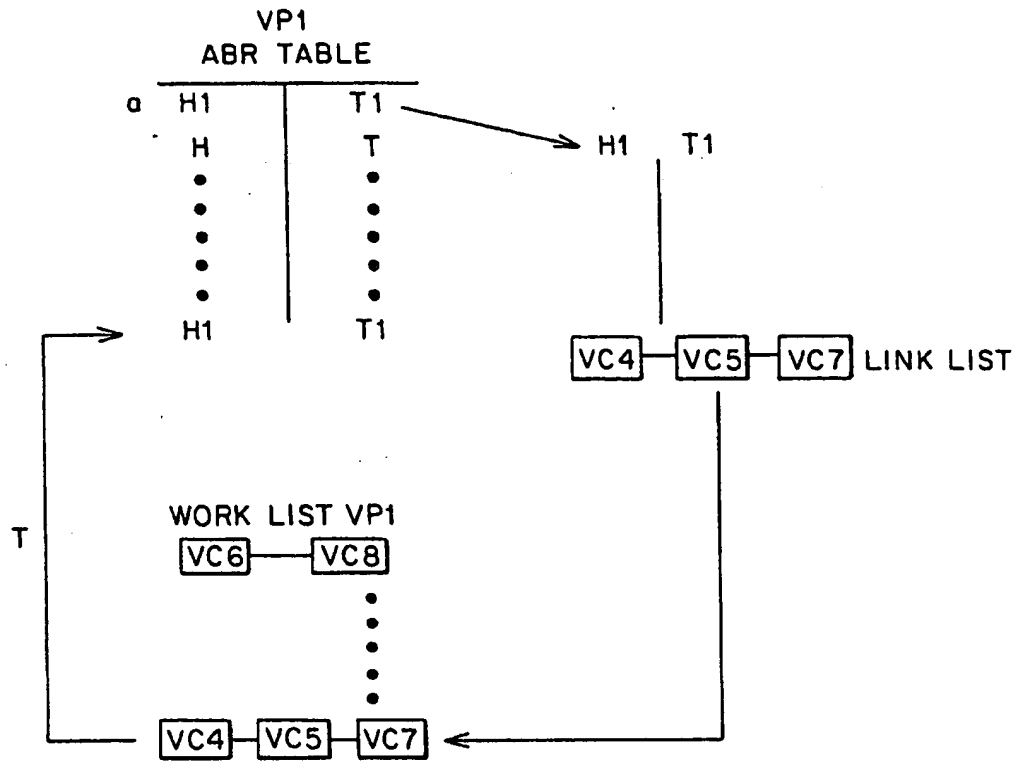
ABR TABLE-VP0

CURRENT
POINTER P0→

ABR HEAD POINTER	ABR TAIL POINTER
H4	T4
H8	T8
•	•
•	•

FIG. 7D

8/9

**FIG. 8**

9/9

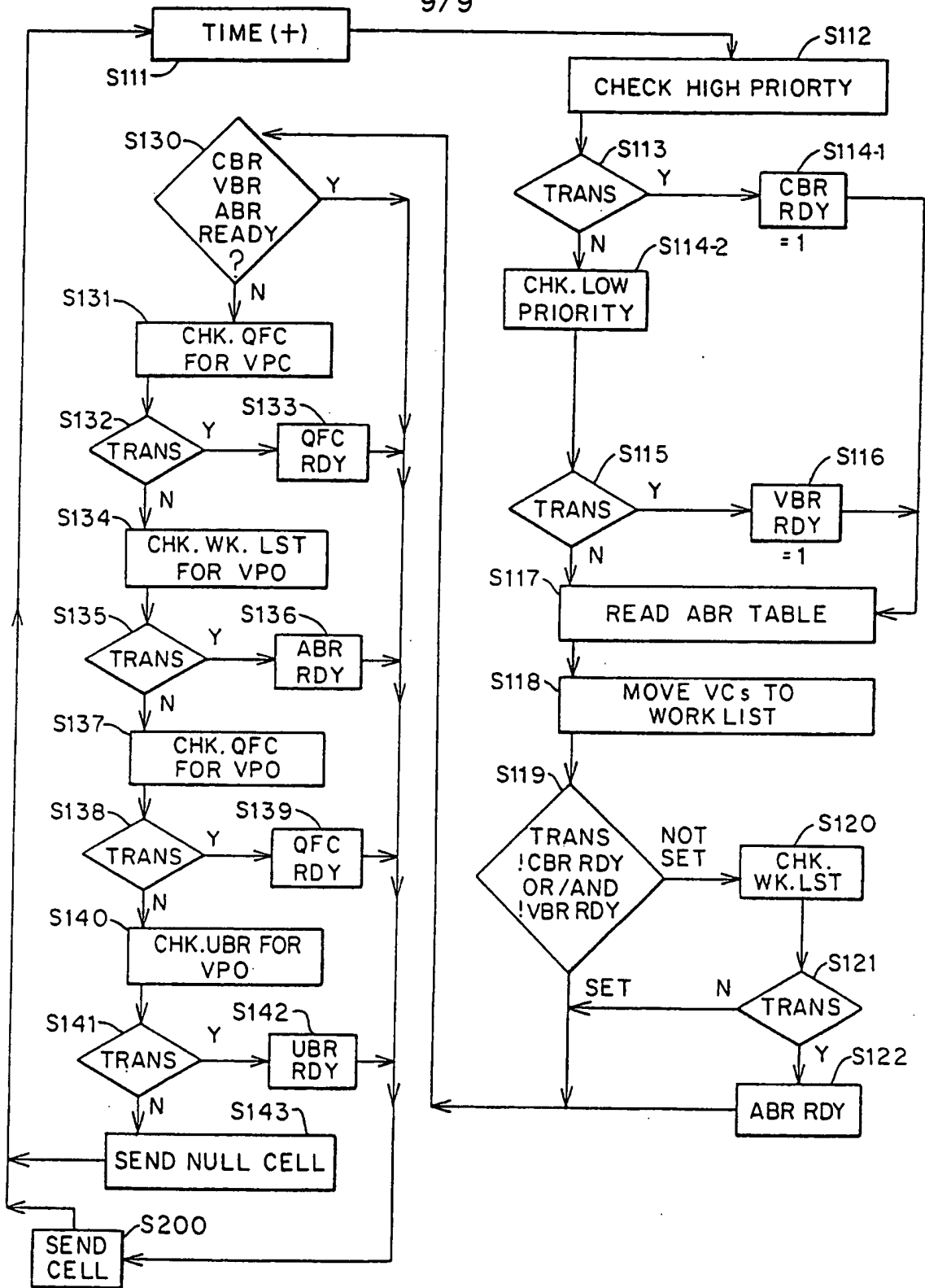


FIG. 9

AMENDED SHEET